N69-36202
NASA CR-105608

PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ON A FUNCTION SPACE APPROACH TO A CLASS
OF LINEAR STOCHASTIC OPTIMAL CONTROL SYSTEMS

J.Y.S. Luh and M.P. Lukas

December, 1968

TR-EE68-36

Lafayette, Indiana 47907

ON A FUNCTION SPACE APPROACH TO A CLASS

OF LINEAR STOCHASTIC OPTIMAL CONTROL SYSTEMS

J.Y.S. Luh and M.P. Lukas

December, 1968

TR-EE68-36

School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

This work was partially supported by the Jet Propulsion Laboratory, California Institute of Technology, Sponsored by the National Aeronautics and Space Administration under Contract NASA 7-100; and by the Joint Service Electronics Program, N0014-67-A-0226.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The stochastic optimal control problem considered in this report is characterized by a dynamic system which is linear in the state and control vectors, and which is disturbed by additive Gaussian white noise. Incomplete, noisy observations of the state vector are available, and the control is required to be a linear feedback function of the estimated state vector. The components of the state vector and control vector which are of interest are lumped together in a response vector, and the performance index to be minimized is then a function of the statistics of the response vector. It is shown that a well-known stochastic control problem, whose performance index is the expected value of a quadratic form on the state and control, is a special case of the more general problem described above.

The general problem is then reformulated as a problem of minimizing a nonlinear functional on a set in a Hilbert space. In this formulation, the well-known "quadratic" problem becomes one of minimizing a linear functional on the same set in the space. Conditions are derived under which the two problems are "equivalent"; that is, the linear and non-linear functionals which specify the problems take on their minimum value at the same point in the space.

A function space algorithm of Dem'yanov is then applied to the

solution of the general problem. This algorithm makes use of the known formal solution to the "quadratic" problem in the iteration procedure. In function space terms, the algorithm iteratively solves the problem of minimizing the nonlinear functional by solving a sequence of linear functional minimization problems.

The above approach is illustrated by two example problems. In the first example, the objective is to find a "minimum variance" control for a third-order dynamic system. In the second example, the objective is to find a control which minimizes wind-gust effects on a large, flexible launch booster. The booster dynamics and wind-gust effects are modeled by a tenth-order time-varying linear differential system. The function space approach and the algorithms developed were found to be useful in obtaining good controls for both examples.

CHAPTER 1

INTRODUCTION

1.1 Optimal Control of Stochastic Systems

As long as control systems have been built and studied, control engineers have had to cope with the presence of noise in these systems. For example, fire control systems in naval vessels are disturbed by thermal noise in the radar subsystem and by the random pitching and rolling motion of the vessel's hull in the sea. A current problem is minimizing the effect of wind-gusts on the trajectories and bending characteristics of large launch boosters. Usually, the practical approach to such problems has been to design the systems conservatively, so that the effects of disturbance noise or sensor noise could be ignored. It is only recently that an organized attack on the problem of noise in control systems has been undertaken, in the form of studies in stochastic stability and stochastic optimal control. As yet, these studies are still in their infancy, and unified results are not plentiful.

The stochastic control theory that has been developed relies heavily on the state variable-differential equation model of dynamic systems. This model can be extended to the case in which random variables are present in the dynamic equations, if the state variables of the system are chosen such that they can be described by a multivariate Markov process (see Wonham, reference [2.7]). Then the stochastic system is described by the joint probability distribution of the state vector components. This distribution can be found by solving a

Kolmogorov partial differential equation, as was done in [2.7]. If the performance index to be minimized is the expected value of a function of the state and control, the imbedding procedure of dynamic programming can be used to derive a type of Hamilton-Jacobi partial differential equation. An expression for the optimal control is found from a minimization operation in the above equation, and this expression is a function of the solution to that partial differential equation. So the optimal control problem is solved if a solution to the Hamilton-Jacobi equation can be found.

The above dynamic programming approach has been the most popular one in stochastic optimal control studies, and has been used by Florentin [1.1], Orford [1.2], Kounias [1.3], and many others. Survey papers on this and other approaches, such as the application of stochastic stability theory and the stochastic maximum principle, have been written by Wonham [2.7], Kushner [1.4,1.5], Paiewonsky [1.6], and Mayne [1.7].

1.2 Motivation of Research

A well-known problem which has been solved by the above dynamic programming approach is one in which the system equations are linear and the performance index is quadratic in the control and state vectors. The plant is disturbed by additive Gaussian white noise, and incomplete, noisy observations are available to the controller. The formal solution to the problem of minimizing the above performance index is known, and is to choose a control which is a linear feedback function of the Kalman filter state estimate (see, e.g., Wonham [2.7]).

In a study of the design of controllers to alleviate wind-gust effects on launch boosters [2.4], Skelton formulated a stochastic control problem similar to the one above, but which had a nonquadratic performance index. This index was a very useful one in practical applications, because it gave an upper bound on the probability that an event of "mission failure" (such as excessive vehicle bending) would occur during the launch. He showed that there were certain similarities between his index and the quadratic one, and conjectured that the two problems could be made to be "equivalent" (i.e., have the same solution) if certain conditions relating the two performance indices were met. He derived necessary conditions for "equivalence" to occur, and also proposed an algorithm for finding the quadratic performance index that was equivalent to his nonquadratic one. Once this index was found, the known solution to the "quadratic" problem was also the solution to his problem.

This concept of "equivalence" of stochastic control problems was an interesting one, but Skelton left many questions unanswered. For example, he gave no conditions that guaranteed the existence of a "quadratic" problem that was equivalent to a nonquadratic one. Also, Skelton's algorithm was not an automatic one, but involved some engineering judgement in the iteration loop, and no proof of convergence of the algorithm was available. Skelton's method was successfully used to obtain good controls in the gust-alleviation problem, however, so it seemed that his approach had much practical merit.

To investigate some of the above concepts in a more rigorous theoretical framework, the problems described above were reformulated as ones of minimizing functionals on a Hilbert space. This formulation turned out to be a fruitful one, because a number of the theoretical results and computational techniques in functional analysis could then be applied to solving Skelton's problem. In particular, a geometric interpretation of Skelton's "equivalence" concept was developed, and conditions which guaranteed the existence of an equivalent "quadratic" problem were derived. Also, a function space algorithm of Dem'yanov's was applied to Skelton's problem, and the algorithm was shown to converge. To illustrate the results obtained, two example problems were solved. In the second example, a suboptimal approach was developed to solve Skelton's booster control problem, which originally motivated the research.

1.3 Organization of the Thesis

The thesis is divided into seven chapters. In Chapter 2, the class of control problems to be considered in the thesis is defined. The formulation is similar to Skelton's in [2.4]. The well-known stochastic control problem mentioned above is shown to be a member of the class, and the formal solution to this problem is given. Chapter 3 reformulates the above problems in a function space, and a geometrical interpretation of Skelton's equivalence concept is given. This chapter also presents a motivation for the equivalence theorem and the algorithm to be developed in later chapters. The main topic of discussion in Chapter 4 is the derivation of a set of conditions which guarantee equivalence between a "quadratic" problem and the more general problem defined in Chapter 2. Chapter 5 gives a function space interpretation of Skelton's algorithm, and introduces the perturbed gradient algorithm. A proof of convergence of the latter algorithm is also given. The computational results of two example problems are given in Chapter 6, to illustrate the usefulness of the methods developed. Conclusions and recommendations for future study are presented in Chapter 7.

CHAPTER 2

A STOCHASTIC OPTIMAL CONTROL PROBLEM

2.1 Introduction

In this chapter, a problem of finding an optimal controller for a linear plant subject to disturbance noise is presented. It is assumed that the plant can be described by a finite number of linear differential equations, and that the (white Gaussian) noise enters additively into the plant equations. It is also assumed that incomplete, noisy observations of the state vector are made, and that the control is a feedback one using these observations. These assumptions are discussed, and a general performance index to be minimized is given.

A special case of this general problem, in which the performance index is a quadratic form in the state and control vectors, is discussed and the well-known formal solution is given.

2.2 Statement of the General Problem

The dynamic system model to be considered is a linear plant described by a differential system and perturbed by an additive white Gaussian disturbance noise:

$$\frac{dx(t)}{dt} = A(t)x(t) + B(t)u + v(t) \ , \qquad (2\text{-}1)$$

with $$x(t_o) = 0 \ , \qquad (2\text{-}2)$$

and $x(t)$ = (n x 1) state vector

u = (m x 1) control vector

$v(t)$ = (n x 1) noise vector.

This particular model is chosen because it can be used to represent many linear physical systems, and can be used as an approximation to certain nonlinear systems about a nominal operating point or trajectory. The assumption of an additive Gaussian noise input to a linear system is a useful one because it guarantees that $x(t)$ is also Gaussian (see Kalman [2.1], Theorem 5). In addition, noise in physical systems can often be approximated by a Gaussian process. The assumption of white noise is not an unduly restrictive one, because "colored noise" can often be represented as the output of a linear filter whose input is white noise. The linear filter equations can then be adjoined to the original system equations, producing a linear system with white additive noise, as originally assumed.

Since not all components of x and u are of interest in the evaluation of performance, an ℓ-dimensional response vector $r(t)$ is defined:

$$r(t) = C(t)x(t) + D(t)u \tag{2-3}$$

It is assumed that incomplete, noisy measurements of the state vector are available:

$$z(t) = H(t)x(t) + w(t) \ , \tag{2-4}$$

where $z(t)$ = (k x 1) measurement vector,

$w(t)$ = (k x 1) noise vector.

Again, the noise $w(t)$ is assumed to be additive, white, and Gaussian for simplicity. The case in which $w(t)$ is "colored" or some of the components of $z(t)$ contain no noise is discussed by Bryson and Johansen [2.2].

The noise vectors $v(t)$ and $w(t)$ are completely described by:

$$E[v(t)] = E[w(t)] = 0 \quad , \tag{2-5}$$

$$E[v(t)v'(\tau)] = N_v(t)\,\delta(t-\tau) \tag{2-6}$$

$$E[w(t)w'(\tau)] = N_w(t)\,\delta(t-\tau) \tag{2-7}$$

$$E[v(t)w'(\tau)] = 0 \quad , \tag{2-8}$$

where $E[\cdot]$ denotes the expectation operator, the prime denotes transpose, and $\delta(t-\tau)$ denotes the Dirac delta function at $t = \tau$.

The following comments should be made:

1) It is assumed that the system operates for a fixed time, $t \in [t_o, T]$, where t_o and T are given.

2) The matrices $A(t)$, $B(t)$, $C(t)$, $D(t)$, $H(t)$, $N_v(t)$, and $N_w(t)$ are all assumed to be known and to have proper dimensions; their elements are assumed to be continuous for $t \in [t_o, T]$.

3) $N_w(t)$ is assumed to be positive definite for all $t \in [t_o, T]$. This assumption is the same as that of no "perfect measurements" mentioned previously.

The set of admissible controls to be considered is:

$$U = \left\{ \begin{array}{l} u : u = -K(t)\hat{x}(t|t), \text{ and the elements of } K(t) \\ \text{are continuous on } [t_o, T] \end{array} \right\}, \tag{2-9}$$

where $K(t) = (m \times n)$ feedback coefficient matrix

$\hat{x}(t|t) = (n \times 1)$ Kalman filter estimate of $x(t)$ given observations $z(\tau)$, $\tau \in [t_o, t)$.

The theory of the Kalman filter is well-established (see, e.g., [2.1] and [2.3]), and is especially useful in the above problem, since x(t) is a Gaussian process. The Kalman filter estimate is then the best estimate, not only in the minimum mean-square error sense, but with respect to other error criteria as well [2.3]. A linear feedback law is assumed in order to guarantee that the control vector is Gaussian; that this is true can be verified by examining the Kalman filter equations. In addition, the linear feedback law is easy to implement in practice, and is therefore useful in applications. The assumption also "decouples" the control problem from the estimation problem, which is now assumed to be solved.

For the above case, the Kalman filter equations are:

$$\frac{d\hat{x}(t|t)}{dt} = [A(t) - B(t)K(t)]\hat{x}(t|t) + E_k(t)H'(t)N_w^{-1}(t)[z(t) - H(t)\hat{x}(t|t)] \,, \tag{2-10}$$

with $\hat{x}(t_o|t_o) = 0$, (2-11)

and where $E_k(t) = E[\tilde{x}(t|t)\tilde{x}'(t|t)]$, (2-12)

and $\tilde{x}(t|t) = x(t) - \hat{x}(t|t)$ (2-13)

$= \text{error vector}.$

The matrix $E_k(t)$ is not a function of the observations z(t), and is the solution of the error covariance equation:

$$\frac{dE_k(t)}{dt} = A(t)E_k(t) + E_k(t)A'(t) - E_k(t)H'(t)N_w^{-1}(t)H(t)E_k(t) + N_v(t) \,, \tag{2-14}$$

with $\quad E_k(t_o) = 0 \quad .$ (2-15)

The other quantities in (2-10) to (2-14) are previously given, so the Kalman filter is completely specified once the coefficient matrix K(t) is given. And so the control u is specified when K(t) is given.

The performance index to be minimized is of the general form:

$$J = f_1[S(T)] + \int_{t_o}^{T} f_2[S(t)]dt \ , \tag{2-16}$$

where $\quad S(t) = E[r(t)r'(t)]$ (2-17)

$\qquad\qquad$ = covariance matrix of the response vector r(t).

Note that S(t) is indeed a covariance matrix, because

$$E[r(t)] = 0 \ , \tag{2-18}$$

which can be easily shown.

Note also that r(t) is a Gaussian process, since all the equations defining r(t) are linear and contain additive Gaussian noise. Since r(t) is a zero-mean process, it is completely described by the covariance matrix S(t). Thus it is not unreasonable to choose a performance index of the above form. Because the characteristics of the process at the terminal time may be of special interest, a separate term involving S(T) is included in the performance index. The process characteristics to be controlled during the time period $[t_o,T)$ are weighted in the integral term.

Using the above definitions, we have the following statement:

<u>General Problem Statement</u>: Choose the control $u \in U$ to minimize the performance index J, subject to the system side-conditions (2-1)

to (2-8) and the Kalman filter side-conditions (2-10) to (2-15).

This problem, using the general performance index in (2-16), has not been solved. A special case of the above problem has been solved, however, and will be discussed in the next section. Skelton in [2.4] also considered a special case; his approach will be discussed in Chapter 3.

2.3 A Special Case: Quadratic Performance Index

Florentin [2.5], Tou [2.6], Wonham [2.7, 2.8], and others have discussed the above problem for the case in which the performance index is the expected value of a quadratic form in the system state and control vectors. In the notation of the general problem, the quadratic performance index is:

$$J_Q = E\left\{r'(T)Q_F(T)r(T) + \int_{t_o}^{T} r'(t)Q(t)r(t)dt\right\}, \qquad (2\text{-}19)$$

where $Q_F(T) = (\ell \times \ell)$ symmetric positive semidefinite matrix with bounded elements,

and $Q(t) = (\ell \times \ell)$ symmetric positive semidefinite matrix whose elements are continuous on $[t_o, T]$.

<u>Note</u>: The matrix $D'(t)Q(t)D(t)$ is required to be positive definite for all $t \in [t_o, T]$ to insure the existence of a solution to the quadratic problem ($D(t)$ is defined in (2-3)).

The performance index J_Q can be rewritten in the general form:

$$J_Q = \mathrm{Tr}\,[Q_F(T)S(T)] + \int_{t_o}^{T} \mathrm{Tr}\,[Q(t)S(t)]dt\ , \qquad (2\text{-}20)$$

where Tr denotes the trace operator (takes the sum of the diagonal elements of a matrix).

The solution to the problem with quadratic performance index (the "quadratic problem") has been found by using the "certainty equivalence principle" as in [2.5] and by the stochastic Hamilton-Jacobi equation of dynamic programming, as in [2.7]. In any case, the optimal controller for the quadratic problem using the notation of the general problem is as follows:

$$u^* = -K^*(t)\hat{x}(t|t) \quad , \tag{2-21}$$

where $\hat{x}(t|t)$ is the Kalman filter estimate of $x(t)$ given observations $z(\tau)$, $\tau\in[t_o,t)$, and is defined by (2-10) to (2-15) using $K^*(t)$ for $K(t)$. The optimal feedback coefficient is given by

$$K^*(t) = [D'(t)Q(t)D(t)]^{-1}[B'(t)P_v(t) + D'(t)Q(t)C(t)] \quad , \tag{2-22}$$

and $P_v(t)$ is the solution of the Riccati equation

$$\begin{aligned}\frac{dP_v(t)}{dt} &= -A'(t)P_v(t) - P_v(t)A(t) - C'(t)Q(t)C(t) \\ &\quad + K^{*\prime}(t)D'(t)Q(t)D(t)K^*(t),\end{aligned} \tag{2-23}$$

with the boundary condition

$$P_v(T) = C'(T)Q_F(T)C(T) \quad . \tag{2-24}$$

It must be noted that the form of the optimal control in (2-21) would be the same if u were only required to be a function of past observations $z(\tau)$, $\tau\in[t_o,t)$. That is, the fact that u^* is a linear feedback law on the Kalman filter state estimate is intrinsic to the quadratic problem, and is not merely a consequence of the requirement that $u \in U$.

The above solution to the problem of minimizing J_Q is generally accepted to be correct, although no completely rigorous proof of the results has been published to date (as far as is known). Since the above results will be used extensively in the following chapters, it is convenient to summarize the solution in the following assertion:

Assertion 2.1 (Solution of "Quadratic Problem"). Suppose we are given:

1) the description of the dynamic system, response vector, and measurement vector in equations (2-1) to (2-4) defined on $[t_o,T]$;
2) the equations (2-5) to (2-8) describing the white Gaussian noise vectors $v(t)$ and $w(t)$;
3) the set of admissible controls given in (2-9);
4) the parameter matrices A, B, C, D, H, N_v, and N_w, with known elements continuous in t on $[t_o,T]$;
5) $N_w(t)$ positive definite for all $t \in [t_o,T]$;
6) the performance index J_Q defined in (2-19), with the associated conditions on $Q_F(T)$, $Q(t)$, and $D'(t)Q(t)D(t)$.

Then the problem of selecting the $u \in U$ such that J_Q is minimized, under the above conditions, has a unique solution, given by (2-21). The optimal feedback coefficient $K^*(t)$ is defined by (2-22) to (2-24), and the Kalman filter state estimate $\hat{x}(t|t)$ is defined by (2-10) to (2-15).

CHAPTER 3

FORMULATION OF THE PROBLEM IN FUNCTION SPACE

3.1 Introduction

In a study of the design of controllers to alleviate wind-gust effects on launch boosters [2.4], Skelton introduced the notion of "quadratic equivalence" into the study of stochastic problems. He formulated the wind-gust problem in the form of the general problem posed in Section 2.2, using a specific form of performance index J. Using an analytic method, he developed necessary conditions that a quadratic problem, as defined in Section 2.3, have the same solution as the more general problem of minimizing J. He assumed that such a quadratic problem exists and that the solution to the general problem exists. The two problems are then said to be "equivalent", since knowing the solution to one implies knowing the solution to the other. A further discussion of the derived necessary conditions is given in Appendix A, and Skelton's algorithm for finding the equivalent quadratic problem is discussed in Section 5.2.

The notion of the "equivalence" of stochastic problems is an interesting one, but Skelton does not give any conditions that guarantee the existence of an equivalent problem. Also, the analytical method he uses does not yield much insight into the meaning of equivalence of control problems. To overcome these difficulties, a geometric interpretation of the problems posed in Chapter 2 was developed, using the theory of minimization of functionals on a Hilbert space. This formulation yields a clear interpretation of equivalence, and suggests conditions on the general problem which guarantee the existence of an equivalent quadratic problem. In addition, algorithms for finding the equivalent problem can be easily visualized using the function space approach.

In this chapter, the stochastic problem is first transformed into a nonlinear deterministic one, so that the equations relating the covariance matrix $S(t)$ to the feedback coefficient $K(t)$ are expressed in a deterministic form. Then the function space σ is defined and its properties derived. The stochastic problems defined in Chapter 2 are then interpreted geometrically in σ, and the notion of equivalence is explained in terms of two functionals taking on their minima at the same point.

3.2 The Stochastic Problem in Deterministic Form

In Chapter 2, the equations which describe the behavior of the system are stochastic in nature. Given a feedback gain coefficient $K(t)$ and the processes $v(t)$ and $w(t)$, the process $x(t)$ is then determined, as is the covariance matrix $S(t)$.

In Appendix B, it is shown that the following set of deterministic equations also determine $S(t)$:

$$\begin{aligned} S(t) = {} & [C(t) - D(t)K(t)]C_x(t)[C'(t) - K'(t)D'(t)] \\ & + D(t)K(t)E_k(t)C'(t) + C(t)E_k(t)K'(t)D'(t) \\ & - D(t)K(t)E_k(t)K'(t)D'(t), \end{aligned} \tag{3-1}$$

where $$C_x(t) = E[x(t)x'(t)] \; , \tag{3-2}$$

and is the solution of:

$$\frac{dC_x(t)}{dt} = [A(t) - B(t)K(t)]C_x(t) + C_x(t)[A'(t) - K'(t)B'(t)] + B(t)K(t)E_k(t) + E_k(t)K'(t)B'(t) + N_v(t) , \tag{3-3}$$

with initial conditions

$$C_x(t_o) = 0 . \tag{3-4}$$

The error covariance matrix $E_k(t)$ was defined in equation (2-12), and satisfies (2-14) and (2-15). The parameters A, B, C, D, K, and N_v have also been defined in Chapter 2. So, by the above equations, S(t) is determined once K(t) and the noise parameters $N_v(t)$ and $N_w(t)$ are specified.

Since S(t) describes the system behavior completely (with respect to the performance index), and is determined once K(t) is given, the equations (3-1) to (3-4) and (2-14) to (2-15) can be regarded as a set of deterministic system equations. Then S(t) is identified as a new "state matrix" of the system, and K(t) as the "control matrix". This method of transforming a stochastic problem into a deterministic one has been used by Jaswinski [3.1], who also derives "state equations" involving the covariance matrix of the original state vector, and using the feedback coefficient matrix as the new control. Also, Kushner [3.2], Mortensen [3.3], and others have converted the linear stochastic system equation into a nonlinear deterministic partial differential equation in the probability density of the state vector.

The admissible control set in this formulation is then:

$$U_K = \left\{ \begin{array}{l} K(t)\text{: the elements of } K(t) \text{ are} \\ \text{continuous on } [t_o, T] \end{array} \right\} . \tag{3-5}$$

This set is, of course, simply a modification of the set U defined in (2-9).

Then we can state the following:

General Deterministic Control Problem: Find the $K(t) \in U_K$ that minimizes the performance index J, subject to the system equations (3-1) to (3-4) and (2-14) to (2-15).

This problem is the same as that posed in Section 2.2, but now the relationship between K(t) and S(t) is brought out more clearly.

3.3 The Function Space σ

In this section, an abstract function space σ will be defined and its properties stated. The interpretation of the control problem in σ will be studied in Section 3.4.

The basic element in σ has the following form:

$$\hat{e} = [e_F, e(t)] \ , \tag{3-6}$$

where e_F = (k x 1) real vector

$e(t)$ = (k x 1) real measurable vector function of t on $[t_o, T]$.

Let $\hat{e} = [e_F, e(t)]$ and $\hat{g} = [g_F, g(t)]$ be two elements in σ. Then the following operations are defined:

a) addition:

$$\hat{e} + \hat{g} \overset{\Delta}{=} [e_F + g_F, e(t) + g(t)] \tag{3-7}$$

b) multiplication by a scalar λ:

$$\lambda\hat{e} \overset{\Delta}{=} [\lambda e_F, \lambda e(t)] \ . \tag{3-8}$$

The null element is defined as:

$$\hat{e} = \theta = [0, 0] . \tag{3-9}$$

For any two vectors $\hat{e}$, $\hat{g} \in \sigma$, an inner product is defined:

$$(\hat{e}, \hat{g}) \triangleq e_F \cdot g_F + \int_{t_o}^{T} e(t) \cdot g(t)dt, \tag{3-10}$$

where the dots indicate the Euclidean scalar product. Define the norm of $\hat{e}$ to be:

$$\|\hat{e}\|_\sigma = (\hat{e}, \hat{e})^{\frac{1}{2}} , \tag{3-11}$$

where the positive square root is chosen.

And let the metric in σ be:

$$\rho(\hat{e}, \hat{g}) \triangleq \|\hat{e} - \hat{g}\|_\sigma \tag{3-12}$$

Then the definition of σ follows:

Definition 3.1. The space σ is the collection of all elements $\hat{e}$ of the form given in (3-6), such that $\|\hat{e}\|_\sigma < \infty$ and the operations (3-7) to (3-12) are defined. Two elements of σ, say $\hat{e}$ and $\hat{g}$, which have the property that $e_F = g_F$ and $e(t) = g(t)$ almost everywhere, are identified as the same element; that is, $\hat{e} = \hat{g}$ if $\rho(\hat{e}, \hat{g}) = 0$.

In Appendix C it is shown that σ is the "direct sum" of a k-dimensional Euclidean space and k L^2-spaces, and is thus a Hilbert space (by Lemma 19, Dunford and Schwarz [3.4], p. 257). The second part of Definition 3.1 is necessary to satisfy the metric space axiom that $\rho(\hat{e}, \hat{g}) = 0$ if and only if $\hat{e} = \hat{g}$. Thus σ is really a space of "equivalence classes" of functions (for a discussion, see Rudin [3.5], pp.65-66).

3.4 The Stochastic Problem Interpreted in σ-space

In Section 3.2, it was shown that the response covariance matrix S(t) could be regarded as a "state" of the deterministic system. For notational convenience, this state matrix will be converted to a state vector s(t) by "stacking" the columns of S(t). That is, if

$$S(t) = \begin{bmatrix} s_{11}(t) & s_{12}(t) & \cdots & s_{\ell 1}(t) \\ s_{21}(t) & s_{22}(t) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_{\ell 1}(t) & \cdots & \cdots & s_{\ell\ell}(t) \end{bmatrix}, \qquad (3\text{-}13)$$

then

$$s(t) = \begin{bmatrix} s_{11}(t) \\ s_{21}(t) \\ \vdots \\ s_{\ell 1}(t) \\ s_{12}(t) \\ \vdots \\ s_{\ell\ell}(t) \end{bmatrix} \qquad (3\text{-}14)$$

is said to be the $(\ell^2 \times 1)$ covariance state vector.

Now, form the element $\hat{s}$:

$$\hat{s} = [s(T), s(t)] \qquad (3\text{-}15)$$

The element $\hat{s}$ is now shown to be a member of the space σ. Consider the equations (3-1) to (3-4) and (2-14), (2-15), which define S(t), and thus also define $\hat{s}$. The matrices C(t) and D(t) are defined to be continuous; K(t) is continuous by (3-5); $C_x(t)$ and $E_k(t)$ are solutions of differential equations and are therefore continuous. So the elements of $S(t)$ are continuous, and therefore measurable and also in L^2. If the dimension of the vectors $s(T)$ and $s(t)$ are identified as $k = \ell^2$, to conform with the notation in Section 3.3, it follows that $\hat{s} \in \sigma$.

The definition below will be needed in the following discussion:

<u>Definition 3.2.</u> The <u>set of attainability</u> $\alpha \subset \sigma$ is defined as follows:

$$\alpha = \left\{ \begin{array}{l} \hat{s} : S(t) \text{ is the solution of the deterministic system} \\ \text{equations, given a } K(t) \in U_K, \text{ for } t \in [t_o, T] \end{array} \right\} .$$

It should be noted that this set differs from the usual set of attainability in that it considers the system response to admissible controls over the whole time interval of interest, not just at some particular terminal time. It can be interpreted as the mapping of U_K into σ by means of the deterministic system equations.

The performance index J of the general problem posed in Chapter 2 can now be interpreted as a nonlinear functional (in general) on $\hat{s} \in \sigma$:

$$J = J(\hat{s}) = f_1[s(T)] + \int_{t_o}^{T} f_2[s(t)]dt \ . \qquad (3\text{-}16)$$

To express the quadratic functional J_Q in similar form, first form the vectors q_F and $q(t)$ from the quadratic coefficient matrices $Q_F(T)$ and $Q(t)$, respectively, using the "stacking" procedure outlined above. Then, referring to equation (2-20), it can be seen that J_Q can be written as:

$$J_Q = J_Q(\hat{s}) = q_F \cdot s(T) + \int_{t_o}^{T} q(t) \cdot s(t)dt \quad . \qquad (3\text{-}17)$$

Thus J_Q is a <u>linear functional</u> on σ.

Using the above notions, we have the following:

General Problem in σ-space: Find the point $\hat{s} \in \alpha$ (and the corresponding $K(t)$) such that the functional $J(\hat{s})$ is minimized on α.

This problem can be visualized geometrically if the following space F is defined:

$$F = \text{product space of } \sigma \text{ and } R^1, \tag{3-18}$$

where R^1 is the real line. Since the values of the functional J are in R^1, the problem of minimizing $J(\hat{s})$ on α can be represented figuratively as shown in Figure 3.1. The set $\alpha \in \sigma$ is shown, along with an arbitrary point $\hat{s} \in \alpha$. The functional $J(\hat{s})$ can be viewed as a hypersurface in F, and $J_Q(\hat{s})$ as a hyperplane. The point $\hat{s}^*$ is the point in α for which $J(\hat{s})$ is a minimum. The matrix $K^*(t)$ which corresponds to $\hat{s}^*$ is then the optimal feedback coefficient, and is the solution to the stochastic control problem. The set α_Q in σ will be defined in Section 3.5.

Other quantities of interest in the discussion to follow are the first and second differentials of the function J (see, e.g., Vainberg [3.6] for definition and discussion), and the gradient vector of J. The explicit expressions for these quantities are given in the theorem below, which is proved in Appendix C.

Theorem 3.1

Assume the following:

1) $J(\hat{s})$ is defined for every $\hat{s} \in \sigma$;
2) f_1, f_2, $\frac{\partial f_1}{\partial s}$, $\frac{\partial f_2}{\partial s}$, $\frac{\partial^2 f_1}{\partial s^2}$, and $\frac{\partial^2 f_2}{\partial s^2}$ exist and are finite for all $t \in [t_o, T]$, and have elements continuous in s for every $\hat{s} \in \sigma$ (see (3-26) and (3-27) for definitions).

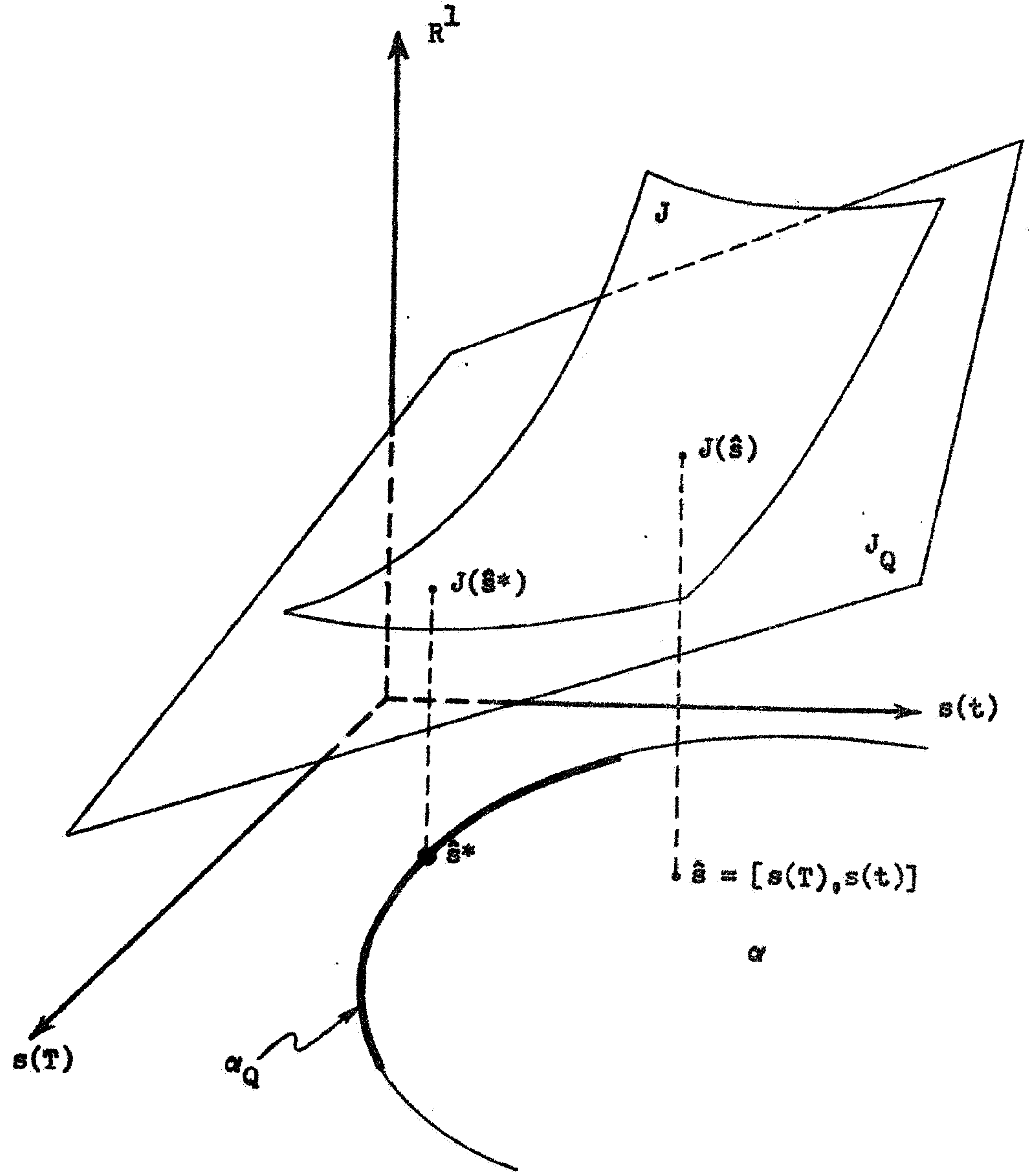


Figure 3.1 The Minimization Problem in F-space

And let $\hat{s} = [e_F, e(t)]$ and $\hat{\eta} = [\eta_F, \eta(t)]$ be arbitrary elements in σ; then (see Appendix D for background materials):

1) J has a Gateaux (weak) differential (see Definition D.1) defined at each $\hat{s} \in \sigma$, for every element $\hat{e}$ in σ, and it is given by:

$$VJ(\hat{s}, \hat{e}) = (DJ(\hat{s}), \hat{e}) , \tag{3-19}$$

where $$DJ(\hat{s}) = \left[\frac{\partial f_1}{\partial s}, \frac{\partial f_2}{\partial s}(t)\right]\Big|_{\hat{s}} \tag{3-20}$$

is the <u>gradient vector</u> of J, and is an element of σ;

2) J has a second Gateaux differential (see Definition D.2) at $\hat{s} \in \sigma$, for all $\hat{e}$, $\hat{\eta} \in \sigma$, given by:

$$V^2J(\hat{s}, \hat{e}, \hat{\eta}) = (D^2J(\hat{s}, \hat{e}), \hat{\eta}) , \tag{3-21}$$

where $$D^2J(\hat{s}, \hat{e}) = \left[\frac{\partial^2 f_1}{\partial s^2} e_F, \frac{\partial^2 f_2}{\partial s^2} e\right]\Big|_{\hat{s}} \tag{3-22}$$

is an element in σ.

3) Further, if $DJ(\hat{s})$ and $D^2J(\hat{s}, \hat{e})$ are continuous in $\hat{s}$ (in the norm of the σ-space), then VJ and V^2J are also continuous in $\hat{s}$.

A corollary of the above theorem follows immediately from the definition of J_Q:

Corollary 3.1

If J_Q is defined as in (3-17), then

$$VJ_Q(\hat{s}, \hat{e}) = (DJ_Q(\hat{s}), \hat{e}) , \tag{3-23}$$

where $$DJ_Q(\hat{s}) = [q_F, q(t)], \tag{3-24}$$

and $$V^2J_Q(\hat{s}, \hat{e}, \hat{\eta}) = 0 . \tag{3-25}$$

The gradient vector defined in the above theorem gives the "direction" in which the functional J rises most rapidly. Since J_Q is a linear functional, DJ_Q is a constant vector and does not depend on $\hat{s}$.

The following notation for the partial derivative vectors and matrices was used above:

$$\frac{\partial f_i}{\partial s} = \begin{bmatrix} \frac{\partial f_i}{\partial s_1} \\ \frac{\partial f_i}{\partial s_2} \\ \vdots \\ \frac{\partial f_i}{\partial s_k} \end{bmatrix}, \quad \frac{\partial^2 f_i}{\partial s^2} = \begin{bmatrix} \frac{\partial^2 f_i}{\partial s_1^2} & \frac{\partial^2 f_i}{\partial s_1 \partial s_2} & \cdots & \frac{\partial^2 f_i}{\partial s_1 \partial s_k} \\ \frac{\partial^2 f_i}{\partial s_2 \partial s_1} & \ddots & & \vdots \\ \vdots & & \ddots & \\ \frac{\partial^2 f_i}{\partial s_k \partial s_1} & \cdots & & \frac{\partial^2 f_i}{\partial s_k^2} \end{bmatrix}, \tag{3-26}$$

for $i = 1,2$, where

$$s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_k \end{bmatrix}, \tag{3-27}$$

and $k = \ell^2$.

As mentioned above, the gradient vector $DJ_Q(\hat{s})$ does not vary with $\hat{s}$. Thus, another interpretation of the problem of minimizing J_Q on α is: find the $\hat{s} \in \alpha$ (and the corresponding $K(t)$) such that $(DJ_Q, \hat{s})$ is minimized. The resulting optimal point, $\hat{s}^*$, is the point in α which is the "farthest" in the direction of the negative gradient vector. Such an optimal point can be shown to exist, if Assertion 2.1 is assumed valid. The assertion guarantees that a unique solution to the J_Q-problem exists, in the form of an optimal feedback coefficient, $K^*(t)$. Using $K^*(t)$ in the deterministic system equations yields $\hat{s}^*$. Since $K^*(t)$ is continuous in t, so is $s^*(t)$, defined by $\hat{s}^* = [s^*(T), s^*(t)]$; and $s^*(T)$ is defined. So $\hat{s}^*$ is an element of α, and is the required optimal point.

The above remarks on minimizing J_Q will be used in the following chapters, and can be summarized by the following theorem:

Theorem 3.2

The stochastic control problem of finding a $u \in U$ to minimize J_Q, outlined in Section 2.3, can be interpreted as finding a point $\hat{s}^*$ in α at which the functional $J_Q(\hat{s})$ takes on its minimum value. Also, such a point $\hat{s}^*$ exists and is unique if the matrices $Q_F(T)$ and $Q(t)$, which define the functional J_Q, satisfy the conditions in (2-19). Further, $\hat{s}^*$ is found by using the optimal feedback coefficient $K^*(t)$, defined in (2-22) to (2-24), in the deterministic system equations (3-1) to (3-4).

3.5 The Equivalence of Stochastic Problems

The method to be used in minimizing the functional $J(\hat{s})$ on α depends on the location of the minimum point $\hat{s}^*$. If $\hat{s}^*$ is known to be in the interior of α, steepest-descent or gradient methods in function space can be used to find $\hat{s}^*$. This problem is essentially that of finding the minimum of a functional on the whole space, and can be attacked in a variety of ways (see, e.g., Kantorovich [3.7], Goldstein [3.8]). The main difficulty is in finding the optimal feedback coefficient $K^*(t)$, given the minimum point $\hat{s}^*$. This is not a trivial problem, due to the nonlinearities in the deterministic system equations (3-1) to (3-4).

The interesting case is that in which $\hat{s}^*$ is known to lie on the boundary of α (assuming that α has a boundary). Note that the point which minimizes J_Q, if such a point exists, must lie on the boundary (to be proved). Also, the method of finding the minimum point is known, since the solution to the "quadratic problem" is known. So, it is conceivable that, under the proper conditions on J, there exists a functional J_Q whose minimizing point $\hat{s}^* \in \alpha$ is also the point which minimizes the functional J. Then the problem of minimizing J_Q is said to be equivalent to that of minimizing J. So if the equivalent problem can be found, its known solution can be used to find the solution to the more general problem posed in Chapter 2.

The conditions on J which will insure the existence of an equivalent problem, and sufficient conditions for two problems to be equivalent are discussed in Chapter 4. In this section, the notion of equivalence is introduced and certain related definitions are made.

For convenience, the problem of finding the point $\hat{s}^* \in \alpha$ at which the functional $J(\hat{s})$ takes on its minimum will be called the "J-problem" (and similarly for J_Q). Now, by definition of the J_Q functional in (3-17), the J_Q-problem is defined when the quadratic coefficients $Q_F(T)$ and $Q(t)$ are given. The following definitions will be used in Chapter 4:

Definition 3.3: A J_Q-problem is said to be admissible if the quadratic coefficients $Q_F(T)$ and $Q(t)$, which define J_Q, satisfy the conditions in (2-19).

Definition 3.4: A point $\hat{s}^* \in \alpha$ is said to be a minimum point of a J-problem if $J(\hat{s}^*) \leq J(\hat{s})$ for all $\hat{s} \in \alpha$.

Definition 3.5: The set $\alpha_Q \subset \alpha$ is the set of minimum points of all admissible J_Q-problems.

The set α_Q is depicted in Figure 3.1. It was previously suggested that the minimum point(s) of a J_Q-problem lie on the boundary of α, so α_Q is shown on the boundary. Note that α_Q is well-defined due to Theorem 3.2.

Definition 3.6: Two problems are said to be equivalent if they have a common minimum point.

It can be seen that, if a minimum point of J lies in α_Q, then an equivalent J_Q-problem exists. The conditions on J to insure this will be discussed in the next chapter. Some methods of actually finding this J_Q-problem are presented in Chapter 5.

The above discussion of equivalence is not intended to be a rigorous one, but is meant to motivate the theorems which will be developed in Chapter 4 and the algorithms to be discussed in Chapter 5. The results in those chapters are a consequence of the function space interpretation of the stochastic problem, and make use of the available theory of the constrained minimization of a functional.

CHAPTER 4

SOLUTION OF THE PROBLEM IN FUNCTION SPACE

4.1 Introduction

In Chapter 2, the stochastic problem to be solved was defined. In Chapter 3 it was interpreted as a problem of minimizing a functional on the σ-space, and the existence of equivalent stochastic problems was conjectured. In this chapter, Theorem 4.1, which gives necessary and sufficient conditions for the equivalence of J- and J_Q-problems, is proved. This is preceded by a preliminary lemma, which guarantees that the functional J (defined in (3-16)) can be expanded in a Taylor series in function space. Then, assuming that α is convex and that the minimum point of J is known, it is shown that an equivalent J_Q-problem exists, and is defined by the gradient vector of J at the minimum point. Conversely, it is also shown that if a J_Q-problem and its solution satisfy certain conditions involving the gradient vector of J, then the solution defines a minimum point of J. The proof of Theorem 4.1 has certain parallels with the proof of Dem'yanov's Theorem 1 in [4.1]. A second theorem, which gives a number of properties of the sets α and α_Q (defined in Chapter 3), is also proved in this chapter. As yet, a general convexity theorem for α is not available. The nonlinearities in the deterministic system equations (see Section 3.2) make it very difficult to derive such a general theorem. However, a method for proving convexity is outlined in Section 4.4, and sufficient conditions for convexity are derived for a simple scalar system. In general, the convexity of σ must be assumed or proved in each particular case if Theorem 4.1 is to be applied to a specific problem. Aside from convexity, however, Theorems 4.1 and 4.2 give a complete set of conditions for solution of the J-problem and for use of the algorithms to be discussed in Chapter 5.

4.2 Equivalence Conditions

In this chapter, it is assumed that a specific J-problem has been posed and must be solved. It was conjectured in Section 3.5 that, if the J-problem met certain conditions, then an equivalent J_Q-problem exists. Then, since the solution to the latter problem is known, so is the solution to the J-problem. The required conditions on J are given in Theorem 4.1.

A preliminary Lemma concerning the Taylor series expansion of J will first be proved:

Lemma 4.1

Assume the following:

1) $J(\hat{s})$ is defined for every $\hat{s} \in \sigma$;

2) f_1, f_2, $\frac{\partial f_1}{\partial s}$, $\frac{\partial f_2}{\partial s}$, $\frac{\partial^2 f_1}{\partial s^2}$, and $\frac{\partial^2 f_2}{\partial s^2}$ exist and are finite for all $t \in [t_o, T]$, and have elements continuous in s for every $\hat{s} \in \sigma$ (see (3-26) and (3-27) for definitions);

3) $DJ(\hat{s})$ and $D^2J(\hat{s}, \hat{s})$ are continuous in $\hat{s}$ in the norm of the σ-space (see (3-20) and (3-22) for definitions).

Then, given $\hat{s}$, $\hat{s} \in \sigma$, the functional J can be expanded in the following ways:

Finite Increment Formula:

$$J(\hat{s} + \gamma(\hat{s} - \hat{s})) = J(\hat{s}) + \gamma(DJ(\hat{s}), \hat{s} - \hat{s}) + o(\gamma) \qquad (4\text{-}1)$$

Lagrange Formula:

$$J(\hat{s} + \gamma(\hat{s} - \hat{s})) = J(\hat{s}) + \gamma(DJ(\hat{s} + \beta(\hat{s} - \hat{s})), \hat{s} - \hat{s}) \qquad (4\text{-}2)$$

Taylor Series:

$$J(\hat{s} + \gamma(\hat{s} - \hat{s})) = J(\hat{s}) + \gamma(DJ(\hat{s}), \hat{s} - \hat{s}) + \frac{\gamma^2}{2}(D^2J(\hat{s} + \beta(\hat{s} - \hat{s}), \hat{s} - \hat{s}), \hat{s} - \hat{s}), \qquad (4\text{-}3)$$

where $$\lim_{\gamma \to 0} \frac{o(\gamma)}{\gamma} = 0, \qquad (4\text{-}4)$$

and γ and β are real constants,

$$\gamma \in [0, 1], \quad \beta \in [0, \gamma]. \qquad (4\text{-}5)$$

Proof

Choose $\hat{s}, \hat{\eta} \in \sigma$, and form the function $g(\gamma)$:

$$g(\gamma) = J(\hat{s} + \gamma\,\hat{\eta}), \quad \gamma \in [0, 1]. \qquad (4\text{-}6)$$

Using (4-6), the derivative of g is defined as:

$$\frac{dg(\gamma)}{d\gamma} = \lim_{\delta \to 0} \frac{J(\hat{s} + \gamma\,\hat{\eta} + \delta\,\hat{\eta}) - J(\hat{s} + \gamma\,\hat{\eta})}{\delta} \qquad (4\text{-}7)$$

By Definition D.1 (in Appendix D), the above expression is simply the Gateaux differential of J at the point $(\hat{s} + \gamma\,\hat{\eta})$, in the direction $\hat{\eta}$. Since the hypotheses of the Lemma are the same as those of Theorem 3.1, the theorem is applicable. Thus it is guaranteed that the differential exists. Use (3-19) in (4-7):

$$\frac{dg(\gamma)}{d\gamma} = \nabla J(\hat{s} + \gamma\,\hat{\eta},\ \hat{\eta}) = (DJ(\hat{s} + \gamma\,\hat{\eta}),\ \hat{\eta}). \qquad (4\text{-}8)$$

Similarly, the second derivative of g is defined as:

$$\frac{d^2 g(\gamma)}{d\gamma^2} = \lim_{\delta \to 0} \frac{\nabla J(\hat{s} + \gamma\,\hat{\eta} + \delta\,\hat{\eta},\ \hat{\eta}) - \nabla J(\hat{s} + \gamma\,\hat{\eta},\ \hat{\eta})}{\delta} \qquad (4\text{-}9)$$

By Definition D.2, (4-9) is the second Gateaux differential of J. And so by (3-21):

$$\begin{aligned} \frac{d^2 g(\gamma)}{d\gamma^2} &= \nabla^2 J(\hat{s} + \gamma\,\hat{\eta},\ \hat{\eta},\ \hat{\eta}) \\ &= (D^2 J(\hat{s} + \gamma\,\hat{\eta},\ \hat{\eta}),\ \hat{\eta}), \end{aligned} \qquad (4\text{-}10)$$

where $D^2J(\hat{s} + \gamma\,\hat{\eta},\ \hat{\eta})$ is defined in (3-22). Now, hypothesis 3) of the Lemma guarantees that the third conclusion of Theorem 3.1 applies; that is, that the differentials ∇J and $\nabla^2 J$ are continuous in $\hat{s}$. By inspection of (4-8) and (4-10), it follows that $g'(\gamma)$ and $g''(\gamma)$ are continuous in γ. So $g(\gamma)$ can be expanded in the following ways, all of which are special cases of the Taylor formula (see, e.g., Kaplan [4.2], p. 357):

$$g(\gamma) = g(0) + \gamma g'(0) + o(\gamma), \qquad (4\text{-}11)$$

$$g(\gamma) = g(0) + \gamma g'(\beta), \qquad (4\text{-}12)$$

$$g(\gamma) = g(0) + \gamma g'(0) + \frac{\gamma^2}{2} g''(\beta), \qquad (4\text{-}13)$$

where $\lim_{\gamma \to 0} \frac{o(\gamma)}{\gamma} = 0$, and $\gamma \in [0,\ 1]$, $\beta \in [0,\ \gamma]$.

Then the Lemma follows by substituting (4-6), (4-8), and (4-10) into (4-11) to (4-13), and letting $\hat{\eta} = \hat{e} - \hat{s}$. Q.E.D.

The following equivalence theorem can then be proved using the results of the above Lemma.

Theorem 4.1

Assume:

1) the set α is convex;

2) a minimum point $\hat{s}^o$ of the J-problem exists;

3) if $\hat{s} \in \alpha$, the matrices $\frac{\partial f_1}{\partial S}$ and $\frac{\partial f_2}{\partial S}(t)$ are positive semidefinite, and $D'(t)\frac{\partial f_2}{\partial S}(t)D(t)$ is positive definite for all $t \in [t_o, T]$, when all the matrices are evaluated at $\hat{s}$ ($D(t)$ is defined in (2-3));

4) the hypotheses of Theorems 3.1 and 3.2 are satisfied.

Then the following results hold:

1) An equivalent J_Q-problem exists, and is specified by:

$$\hat{q} = DJ(\hat{s}^o) \; ; \qquad (4\text{-}14)$$

that is, $J_Q(\hat{s}^o) \leq J_Q(\hat{s})$ and $J(\hat{s}^o) \leq J(\hat{s})$ for all $\hat{s} \in \alpha$, where $J_Q(\hat{s}) = (DJ(\hat{s}^o), \hat{s})$.

2) Assume, in addition, that:

a) $J(\hat{s})$ is a convex functional;

b) A point $\hat{s}^+ \in \alpha$ is found such that it is a minimum point of the J_Q-problem defined by $\hat{q} = DJ(\hat{s}^+)$; i.e., $\hat{q}$ can be computed from $DJ(\hat{s}^+)$.

Then $\hat{s}^+$ is also a minimum point of the J-problem (and so by conclusion 1) above, the J-problem and the J_Q-problem which satisfies the relation $\hat{q}=DJ(\hat{s}^+)$ are equivalent).

Proof

1) By hypothesis, a minimum point of J exists. Let $\hat{s}^o$ be such a point; that is,

$$J(\hat{s}^o) \leq J(\hat{s}) \qquad \forall\ \hat{s} \in \alpha \ . \tag{4-15}$$

Now, consider the J_Q-problem defined by $\hat{q} = DJ(\hat{s}^o)$. Using Definition 3.3, this J_Q-problem is admissible by hypothesis 3. Therefore, by Theorem 3.2, a minimum point of J_Q exists. Let $\hat{s}^*$ be such a point; that is,

$$J_Q(\hat{s}^*) \leq J_Q(\hat{s}) \qquad \forall\ \hat{s} \in \alpha \ . \tag{4-16}$$

Following an argument of Dem'yanov and Rubinov in [4.1], it will be shown that

$$J_Q(\hat{s}^*) = J_Q(\hat{s}^o) \ . \tag{4-17}$$

Hypothesis 4) indicates that the assumptions of Theorem 3.1 are satisfied; these assumptions are the same as the hypotheses of Lemma 4.1; so the Lemma is valid. Using the finite increment formula in this Lemma, it follows that:

$$J(\hat{s}^o + \gamma(\hat{s}^* - \hat{s}^o)) - J(\hat{s}^o) = \gamma(DJ(\hat{s}^o),\ \hat{s}^* - \hat{s}^o) + o(\gamma) \ , \tag{4-18}$$

where $\lim\limits_{\gamma \to 0} \dfrac{o(\gamma)}{\gamma} = 0.$

The convexity assumption on α insures that $\hat{s}^o + \gamma(\hat{s}^* - \hat{s}^o)$ is in α. Then, since J is minimized at $\hat{s}^o$, the left side of (4-18) is non-negative for all $\gamma \in [0, 1]$. When γ is small, the sign of the right side of (4-18) is determined by the first term; so

$$(DJ(\hat{s}^o),\ \hat{s}^* - \hat{s}^o) \geq 0 \ , \tag{4-19}$$

or

$$(DJ(\hat{s}^o),\ \hat{s}^*) \geq (DJ(\hat{s}^o),\ \hat{s}^o) \ . \tag{4-20}$$

Using the definition of J_Q to rewrite (4-20), we have:

$$J_Q(\hat{s}^*) \geq J_Q(\hat{s}^o) . \tag{4-21}$$

Combining (4-21) with (4-16) yields (4-17).

By Theorem 3.2, the point $\hat{s}^*$ which minimizes J_Q is unique; so $\hat{s}^* = \hat{s}^o$. From (4-15),

$$J(\hat{s}^*) \leq J(\hat{s}) \qquad \forall\ \hat{s} \in \alpha . \tag{4-22}$$

Thus $\hat{s}^*$ is a minimum point of J, and the J_Q-problem specified by $\hat{q} = DJ(\hat{s}^*) = DJ(\hat{s}^o)$ is equivalent to the J-problem by Definition 3.6. The proof of part 1) of the theorem is thus complete.

2) By the additional given assumptions of part 2), a point $\hat{s}^+ \in \alpha$ exists such that:

$$J_Q(\hat{s}^+) = (DJ(\hat{s}^+),\ \hat{s}^+) \leq (DJ(\hat{s}^+),\ \hat{s}) = J_Q(\hat{s}) \tag{4-23}$$

for all $\hat{s} \in \alpha$. Part 2 of the theorem will be proved by contradiction, following an argument of Dem'yanov and Rubinov in [4.1].

Suppose that $\hat{s}^+$ is <u>not</u> a minimum point of J. That is, a point $\hat{e} \in \alpha$ exists such that

$$J(\hat{e}) < J(\hat{s}^+) . \tag{4-24}$$

Using the Lagrange formula in Lemma 4.1 with $\gamma = 1$ and $\hat{s} = \hat{s}^+$, we have:

$$J(\hat{e}) - J(\hat{s}^+) = (DJ(\hat{s}^+ + \beta(\hat{e} - \hat{s}^+)),\ \hat{e} - \hat{s}^+) , \tag{4-25}$$

with $\beta \in [0, 1]$. Form the function $g(\beta)$:

$$g(\beta) = J(\hat{s}^+ + \beta(\hat{e} - \hat{s}^+)) . \tag{4-26}$$

Then, by equation (4-8) in the proof of Lemma 4.1, the derivative $g'(\beta) = \frac{dg(\beta)}{d\beta}$ exists and is given by the right side of (4-25). But $g(\beta)$ is a convex function of β, since J was assumed to be a convex functional. So $g'(\beta)$ is monotone nondecreasing in β, and

$$g'(\beta) \geq g'(0) \; . \tag{4-27}$$

Using (4-27) in (4-25) results in:

$$J(\hat{s}) - J(\hat{s}^{+}) \geq (DJ(\hat{s}^{+}),\, \hat{s} - \hat{s}^{+}) \; . \tag{4-28}$$

Combining (4-24) with (4-28) yields

$$(DJ(\hat{s}^{+}),\, \hat{s} - \hat{s}^{+}) < 0 \; , \tag{4-29}$$

or

$$(DJ(\hat{s}^{+}),\, \hat{s}) < (DJ(\hat{s}^{+}),\, \hat{s}^{+}) \; . \tag{4-30}$$

But this is a contradiction of (4-23), and so $\hat{s}^{+}$ must be a minimum point of J. This proves the second part of the theorem, and the proof of Theorem 4.1 is complete. Q.E.D.

Theorem 4.1 is useful in that it gives conditions on a J-problem that insure the existence of an equivalent J_Q-problem. If these conditions are satisfied, the algorithms described in Chapter 5 for actually finding the equivalent problem can be applied. Then, if a point $\hat{s}^{+} \in \alpha$ is found using the above computational methods, and satisfies the conditions in Part 2) of the theorem, it is the desired solution to the J-problem.

4.3 Properties of α and α_Q

In this section, certain properties of the sets α and α_Q (defined in Chapter 3) are derived. These properties are useful in determining whether a particular J-problem satisfies the hypotheses of Theorem 4.1, and also yield additional insight into the nature of the J- and J_Q-problems. The results are summarized in the following theorem:

Theorem 4.2

If α and α_Q are as defined in Chapter 3, and if, for every $K(t) \in U_K$, the response vector $r(t)$ is a random process with finite and nonzero variance, then:

1) the null element $\theta \notin \alpha$;

2) if $\hat{s} \in \alpha$, the corresponding covariance matrix S is positive semidefinite;

3) α is contained in a half space in σ;

4) at every point $\hat{s}^* \in \alpha_Q$, a supporting hyperplane to α exists;

5) α_Q is on the boundary of α.

Proof

1) The only way that the null element θ could be in α is if an element $\hat{s}_\theta = [s_\theta(T), s_\theta(t)] \in \alpha$ would exist, such that $s_\theta(T) = 0$ and $s_\theta(t) = 0$ for all $t \in [t_o, T]$. But this is impossible, since it was assumed that the random vector $r(t)$ is a process with zero mean and a nonzero variance for every $K(t) \in U_K$. So $\theta \notin \alpha$, as was to be proven. This result simply means that a trivial response vector (identically zero) is excluded from consideration.

2) This statement follows from the well-known fact that any covariance matrix is positive semidefinite (see, e.g., Gnedenko [4.3], p. 200).

3), 4), 5) The last three results follow directly from the interpretation of the J_Q-problem given in Theorem 3.2. Pick a point $\hat{s}^* \in \alpha_Q$, and consider the J_Q-problem which yields that $\hat{s}^*$, and is defined by $\hat{q} = [q_F, q(t)]$. That is,

$$J_Q(\hat{s}^*) = (\hat{q}, \hat{s}^*) \leq (\hat{q}, \hat{s}) = J_Q(\hat{s}) \ \forall \hat{s} \in \alpha. \qquad (4\text{-}31)$$

The following two definitions are needed to continue the proof.

Consider the representation of the J_Q-problem shown in Figure 4.1. Define the hyperplane $L(\hat{q})$ in the following way:

<u>Definition 4.1</u>: A point $\hat{e}$ is in $L(\hat{q})$ if and only if $(\hat{q}, \hat{s} - \hat{s}^*) = 0$.

Then $L(\hat{q})$ divides σ into two half-spaces, σ^+ and σ^-, defined as follows:

<u>Definition 4.2</u>: A point $\hat{s}$ is in σ^+ if and only if $(\hat{q}, \hat{e} - \hat{s}^*) \geq 0$, and $\hat{e}$ is in σ^- if and only if $(\hat{q}, \hat{s} - \hat{s}^*) < 0$.

But now by (4-31), if $\hat{s} \in \alpha$, then $(\hat{q}, \hat{s} - \hat{s}^*) \geq 0$; so $\hat{s} \in \sigma^+$. Thus $\alpha \subset \sigma^+$, and part 3) of the theorem is proved. Also, $L(\hat{q})$ is a supporting hyperplane to α at $\hat{s}^*$, since $\hat{s}^*$ is clearly in $L(\hat{q})$ and $\alpha \subset \sigma^+$. So part 4) of the theorem is proved.

To show that α_Q is on the boundary of α, it must be shown that, given a point $\hat{s} \in \alpha_Q$, every neighborhood of $\hat{s}$ contains a point not in α. Consider the point $\hat{s}^*$ mentioned before, and define the following β-neighborhood of $\hat{s}^*$:

$$N_\beta(\hat{s}^*) = \{\hat{e} : \hat{e} \in \sigma, \text{ and } \|\hat{e} - \hat{s}^*\|_\sigma < \beta\} \qquad (4\text{-}32)$$

Choose a $\beta > 0$, and consider the point $\hat{s}_\gamma$ (where it is assumed that $\|\hat{q}\|_\sigma > 0$):

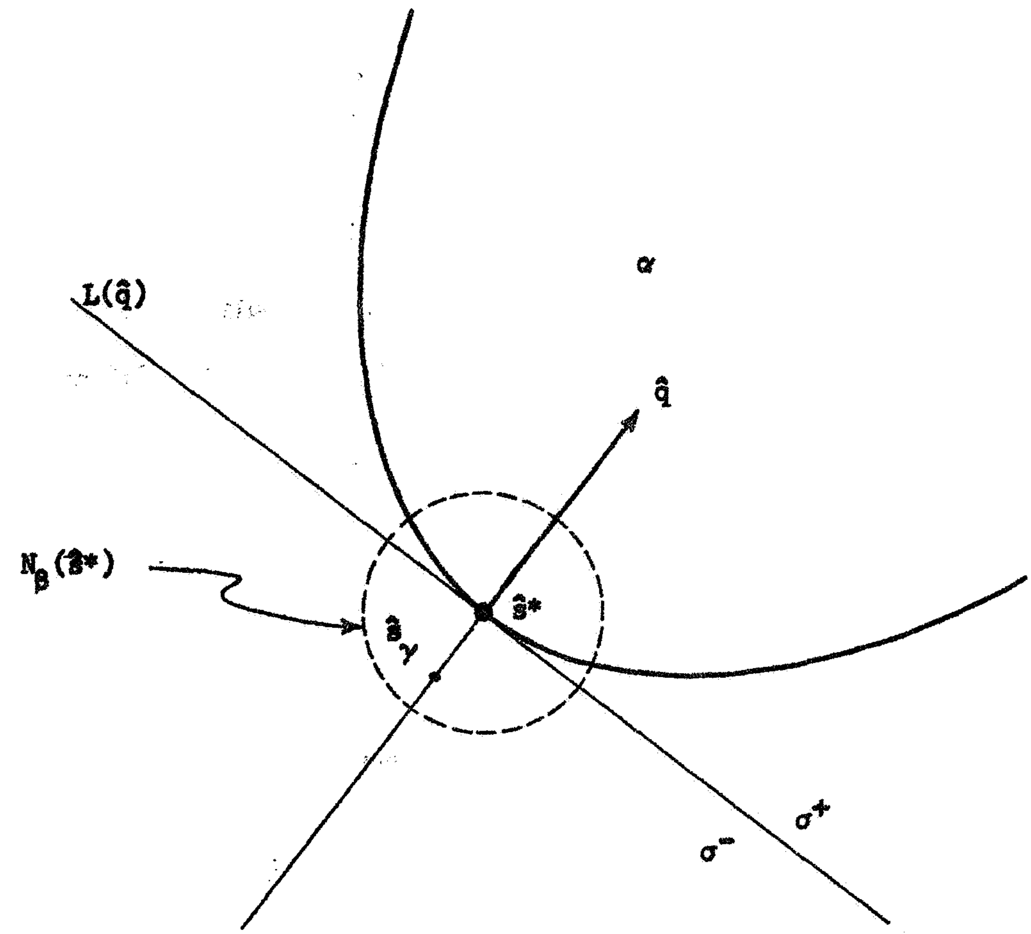


Figure 4.1 Properties of α and α_Q

$$\hat{s}_\gamma = \hat{s}^* - \frac{\gamma\,\hat{q}}{\|\hat{q}\|_\sigma}\,, \text{ for } \gamma \in (0,\,\beta)\,. \tag{4-33}$$

Then $\hat{s}_\gamma \in \sigma$, and

$$\|\hat{s}_\gamma - \hat{s}^*\|_\sigma = \frac{\|\gamma\,\hat{q}\|_\sigma}{\|\hat{q}\|_\sigma} = \gamma\,; \tag{4-34}$$

so $\hat{s}_\gamma \in N_\beta(\hat{s}^*)$. But

$$\begin{aligned}(\hat{q},\,\hat{s}_\gamma - \hat{s}^*) &= (\hat{q},\, -\frac{\gamma\,\hat{q}}{\|\hat{q}\|_\sigma}) \\ &= -\gamma\,\|\hat{q}\|_\sigma\end{aligned} \tag{4-35}$$

by the definition of the norm in (3-11). Since $\gamma > 0$ and $\|\hat{q}\|_\sigma > 0$, $(\hat{q},\,\hat{s}_\gamma - \hat{s}^*)$ is negative, and so $\hat{s}_\gamma$ is in σ^- by Definition 4.2. Since $\alpha \subset \sigma^+$, it follows that $\hat{s}_\gamma \notin \alpha$. The above construction can be carried out for all $\beta > 0$, and so $\hat{s}^*$ is on the boundary of α. This completes the proof of part 5) of the Theorem and thus of the complete Theorem 4.2. Q.E.D.

4.4 Convexity of α

An approach to determining the convexity of α is discussed in this section. In this discussion, let α be defined as follows:

$$\alpha = \left\{\begin{array}{l}\hat{s}{:}S(t) \text{ is the solution of the deterministic system} \\ \text{equations (3-1) to (3-4), given a } K(t) \in \bar{U}_K, \text{ for} \\ t \in [t_o, T].\end{array}\right\}, \tag{4-36}$$

which is similar to the definition of α in section 3.4, except that the set of admissible feedback coefficients is now:

$$\bar{U}_K = \left\{\begin{array}{l}K(t){:}\text{the elements of } K(t) \text{ have continuous} \\ \text{first derivatives on } [t_o, T].\end{array}\right\}. \tag{4-37}$$

Consider the special case in which the A, B, C, and D matrices in (3-1) to (3-4) are constant, and the measurements of the state vector are exact, so that $E_x(t) \equiv 0$ for all t. Then the system covariance equations become:

$$S(t) = [C - DK(t)]C_x(t)[C' - K'(t)D'] \;, \tag{4-38}$$

where $C_x(t)$ is the solution of:

$$\frac{dC_x(t)}{dt} = [A - BK(t)]C_x(t) + C_x(t)[A' - K'(t)B'] + N_v(t) \;, \tag{4-39}$$

with initial conditions

$$C_x(t_o) = C_{xo} \;. \tag{4-40}$$

(Note that here $x(t_o) = x_o$ is a Gaussian random vector with zero mean and covariance matrix C_{xo}, which is assumed to be nonsingular).

To show that α is convex, first choose arbitrary elements $\hat{s}_1$ and $\hat{s}_2$ from α, and form the element $\hat{s}_\lambda$:

$$\hat{s}_\lambda = (1 - \lambda)\,\hat{s}_1 + \lambda\,\hat{s}_2 \;, \qquad \lambda \in (0, 1).$$

From the definition of $\hat{s}$ in Section 3.4, it follows that the covariance matrix corresponding to $\hat{s}_\lambda$ is given by:

$$S_\lambda(t) = (1 - \lambda)\,S_1(t) + \lambda\,S_2(t) \;, \tag{4-41}$$

where $S_1(t)$ and $S_2(t)$ correspond to $\hat{s}_1$ and $\hat{s}_2$. For convexity, it must then be shown that $\hat{s}_\lambda$ is in α. From the correspondence between $\hat{s}_\lambda$ and $S_\lambda(t)$, this is true if and only if there exists a feedback coefficient

$K_\lambda(t) \in \bar{U}_K$, such that $K_\lambda(t)$ produces $S_\lambda(t)$ when used in equations (4-38) to (4-40).

The proof of the existence of such a $K_\lambda(t)$ is nontrivial, because equations (4-38) to (4-40) are nonlinear in K. One method of proof begins by constructing a differential equation for S(t). This can be done by differentiating (4-38) and substituting (4-40) and (4-39) in the result (assuming that [C - DK(t)] is square and nonsingular), yielding:

$$\begin{aligned}\frac{dS(t)}{dt} = &- S(t)[C' - K'(t)D']^{-1}\,\dot{K}'(t)D' \\ &- D\dot{K}(t)[C - DK(t)]^{-1}\,S(t) \\ &+ [C - DK(t)][A - BK(t)][C - DK(t)]^{-1}\,S(t) \qquad (4\text{-}42) \\ &+ S(t)[C' - K'(t)D']^{-1}[A' - K'(t)B'][C' - K'(t)D'] \\ &+ [C - DK(t)]N_v(t)[C' - K'(t)D'] \ ,\end{aligned}$$

where $S(t_o) = [C - DK(t_o)]C_{xo}[C' - K'(t_o)D']$. (4-43)

Assume that $\hat{s}_1$ and $\hat{s}_2$ have been chosen from α. Then the properties of $S_\lambda(t)$ and $\frac{dS_\lambda(t)}{dt}$ are well-defined through equation (4-41). Suppose now that equation (4-42) can be solved for $\dot{K}(t)$ to form the following differential equation:

$$\frac{dK(t)}{dt} = F[K(t),\ S(t),\ \dot{S}(t)] \ , \qquad (4\text{-}44)$$

with $\quad K(t_o) = K_o \qquad (4\text{-}45)$

defined by (4-43). Then, if a $K_\lambda(t)$ exists which produces $S_\lambda(t)$, it will be defined by (4-44) when S_λ and $\dot{S}_\lambda$ are substituted:

$$\frac{dK_\lambda(t)}{dt} = F[K_\lambda(t), S_\lambda(t), \dot{S}_\lambda(t)] \; , \tag{4-46}$$

with $$K_\lambda(t_o) = K_{\lambda o} \; . \tag{4-47}$$

The proof of convexity of σ then reduces to the problem of showing that a solution to (4-46) exists and is in $\bar{U}_K$, given the properties of S_λ and $\dot{S}_\lambda$. The following scalar example gives sufficient conditions that the $\hat{s}_\lambda$ defined by (4-41) is an element of σ, given two points in σ, $\hat{s}_1$ and $\hat{s}_2$, which satisfy given conditions.

Example of Convexity Proof

Consider the scalar dynamic system with state x and control u:

$$\frac{dx(t)}{dt} = x(t) + u(t) \; , \tag{4-48}$$

with $$x(t_o) = x_o \; , \tag{4-49}$$

and $$E[x_o] = 0 \; . \tag{4-50}$$

The response is

$$r(t) = u(t) \; , \tag{4-51}$$

and let the noise variance $n_v(t) = 1$.

Under the above assumptions, the system A, B, and D matrices reduce to unity, and C = 0 (refer to the general system equations in Section 2.2). Therefore, (4-38) to (4-40) become:

$$s(t) = k^2(t)\, c_x(t) \tag{4-52}$$

$$\dot{c}_x(t) = 2[1 - k(t)]c_x(t) + 1 \; , \tag{4-53}$$

with $$c_x(t_o) = c_{xo} = E[x_o^2] \ . \tag{4-54}$$

Following the method outlined above, a differential equation for $s(t)$ given $k(t)$ will be constructed. First, differentiate (4-52):

$$\dot{s}(t) = k^2(t)\dot{c}_x(t) + 2k(t)\dot{k}(t)c_x(t) \ . \tag{4-55}$$

Then multiply both sides of (4-55) by $k(t)$ and substitute (4-52) and (4-53) to eliminate c_x:

$$k(t)\,\frac{ds(t)}{dt} = 2[k(t) - k^2(t) + \dot{k}(t)]s(t) + k^3(t), \tag{4-56}$$

with $$s(t_o) = k^2(t_o)\ c_{xo} \ . \tag{4-57}$$

Note that if $k(t) = 0$ for some t, $s(t)$ is not defined by (4-56); but then $s(t) = 0$ by (4-52).

Now, choose $\hat{s}_1$ and $\hat{s}_2$ from α; then the corresponding time functions are $s_1(t)$ and $s_2(t)$. Form $s_\lambda(t)$:

$$s_\lambda(t) = (1 - \lambda)\ s_1(t) + \lambda\ s_2(t), \quad \lambda \in (0, 1) \ . \tag{4-58}$$

Then, if a k_λ which produces s_λ exists, it must be defined by the following differential equation:

$$k_\lambda(t)\,\frac{ds_\lambda(t)}{dt} = 2[k_\lambda(t) - k_\lambda^{\,2}(t) + \dot{k}_\lambda(t)]s_\lambda(t) + k_\lambda^{\,3}(t), \tag{4-59}$$

with $$s_\lambda(t_o) = k_\lambda^{\,2}(t_o)\ c_{xo} \ . \tag{4-60}$$

The functions $s_\lambda(t)$ and $\dfrac{ds_\lambda(t)}{dt}$ are well defined by (4-58) once λ is chosen. So (4-59) can be rewritten to define k_λ more explicitly:

$$\frac{dk_\lambda(t)}{dt} = \beta(t)k_\lambda(t) + k_\lambda^{\,2}(t) + \gamma(t)k_\lambda^{\,3}(t) \ , \tag{4-61}$$

where $$k_\lambda(t_o) = \left[\frac{s_\lambda(t_o)}{c_{xo}}\right]^{\frac{1}{2}} = k_{\lambda o} , \tag{4-62}$$

and $$\beta(t) = \frac{\dot{s}_\lambda(t) - 2a_\lambda(t)}{2s_\lambda(t)} , \gamma(t) = - \frac{1}{2s_\lambda(t)} . \tag{4-63}$$

The hypothesis of Theorem 4.2 is assumed; namely, that the response $r(t)$ has a nonzero variance for all $t \in [t_o, T]$. Thus $s_\lambda(t) > 0$, and $\beta(t)$ and $\gamma(t)$ in (4-63) exist.

To find the conditions under which a solution to (4-61) exists on $[t_o, T]$, the Cauchy-Peano existence theorem will be used (see, e.g., Coddington and Levinson [4.4], p. 6):

Theorem (Cauchy-Peano)

Consider the differential equation:

$$\left.\begin{aligned} &\frac{dx}{dt} = f(t, x) , \\ \text{where} \quad &x(t_o) = x_o . \end{aligned}\right\} \tag{E}$$

If f is continuous in t and x on the rectangle R (defined by $|t - t_o| \leq a$, $|x - x_o| \leq b$, with $a, b > 0$), then there exists a solution $\varphi \in C^1$ of (E) on $|t - t_o| \leq I$, for which $\varphi(t_o) = x_o$ ($I = \min[a, b/M]$, where $M = \max|f(t, x)|$ on R).

Applying the above theorem to (4-61), we see that f is continuous in t and k_λ for $|t - t_o| \leq T$, $|k_\lambda - k_{\lambda o}| \leq b$, for all $b > 0$. For a given b, we have from (4-61):

$$\begin{aligned} |f(t, k_\lambda)| &= |\beta(t)k_\lambda(t) + k_\lambda{}^2(t) + \gamma(t)k_\lambda{}^3(t)| \\ &\leq \beta_m|k_\lambda(t)| + |k_\lambda(t)|^2 + \gamma_m|k_\lambda(t)|^3 , \end{aligned} \tag{4-64}$$

where $$\beta_m = \max_{t\in[t_o,T]} |\beta(t)| \,, \tag{4-65}$$

and $$\gamma_m = \max_{t\in[t_o,T]} |\gamma(t)| \,. \tag{4-66}$$

Note that β_m and γ_m exist, because $s_\lambda(t)$ and $\dot{s}_\lambda(t)$ are given time functions continuous on $[t_o,T]$. Also, since $|k_\lambda - k_{\lambda o}| \leq b$, and $k_{\lambda o} > 0$, it follows that $|k_\lambda| \leq b + k_{\lambda o}$. So (4-64) becomes:

$$\begin{aligned} |f(t,k_\lambda)| &\leq \beta_m(b + k_{\lambda o}) + (b + k_{\lambda o})^2 + \gamma_m(b + k_{\lambda o})^3 \\ &= M(b) \,, \end{aligned} \tag{4-67}$$

where M(b) is now the upper bound mentioned in the above Theorem (given a particular b). Now, form

$$g(b) = \frac{b}{M(b)} \,. \tag{4-68}$$

Then, by the theorem, if a "b" exists such that $g(b) \geq T$, the solution of (4-61) exists over $[t_o,T]$. To find such a "b", assume that s_λ and $\dot{s}_\lambda$ are such that $\beta_m = \gamma_m = k_{\lambda o} = 1$. Differentiate (4-68) and set the result equal to zero to find $b \geq 0$ such that g(b) is a maximum:

$$M(b) - bM'(b) = 0 \,. \tag{4-69}$$

Rewriting (4-69) and using the given numbers results in:

$$2b^3 + 4b^2 - 3 = 0 \,,$$

which has a real root of $b = 0.740$ (the other two roots are imaginary). For this value of b, $g(b) = 0.0739$. Therefore, by the Cauchy-Peano Theorem, a solution $k_\lambda(t)$ to (4-61) exists for all $t\in[t_o,t_o + 0.0739]$. So if $T \leq t_o + 0.0739$, the solution exists over the whole interval of interest. Furthermore, then $k_\lambda(t)$ has a continuous first derivative, and so it is a member of $\overline{U}_K$ by (4-37). Thus the $\hat{s}_\lambda$ which results from using k_λ in the system equations is a member of α.

It should be noted that the result holds for all s_λ such that $\beta_m = \gamma_m = k_{\lambda o} = 1$. That is, if $\hat{s}_1$, $\hat{s}_2 \in \alpha$ yield an s_λ with the above properties, the "line" joining $\hat{s}_1$ and $\hat{s}_2$ is also in α. So the above example demonstrates the method of proving convexity outlined previously, and shows convexity for the portion of α satisfying the above conditions on $\hat{s}_1$ and $\hat{s}_2$.

CHAPTER 5

COMPUTATIONAL ALGORITHMS

5.1 Introduction

In Chapters 3 and 4 it was shown that the problem of minimizing the performance index $J(\hat{s})$, $\hat{s} \in \sigma$, could be viewed as a problem of minimizing a nonlinear functional on a "set of attainability" α in the Hilbert space σ (see the statement of the General Problem in Section 3.4). It was also shown that α is not the whole space σ, and that a linear functional $J_Q(\hat{s})$ "equivalent" to $J(\hat{s})$ existed under certain conditions on J. In this chapter, two algorithms, the perturbed gradient method (PGM), and the direct gradient iteration method (DGIM), for minimizing the J_Q-functional will be described.

The problem of minimizing a functional on a constraint set in function space has been discussed by other authors. Blum, for example, considers in [5.1] the minimization of a functional subject to equality constraints. Balakrishnan [5.2] considers a special type of minimum-norm problem, under a control energy constraint, using a steepest-descent method. In both of the above problems, it is assumed that an explicit expression for the constraint equation is known.

The algorithms discussed in this chapter differ from the above methods in two ways. First, no explicit expression for the constraint set α is required in the DGIM and PGM algorithms. Second, the objective of the iteration methods is to find the equivalent J_Q-problem. This problem then defines the minimum point of J. Another feature of the algorithms discussed is that they make use of the known solution of the problem of minimizing the linear functional J_Q.

In the following discussion, it is assumed that the hypotheses of Theorems 3.1, 3.2, and 4.1 are satisfied. Additional hypotheses will be required to show convergence of the PGM algorithm, and these are listed in Theorem 5.1.

5.2 Direct Gradient Iteration Method (DGIM)

The direct gradient iteration method of minimizing $J(\hat{s})$ was devised by Skelton in [2.4]. The method was not viewed by Skelton as one in function space, but this interpretation is useful to relate DGIM to the other algorithm to be discussed in Section 5.3.

A block diagram describing DGIM is shown in Figure 5.1. The notation is the same as that in Chapter 3: the vector $\hat{q}_i \in \sigma$ defines a J_Q-problem; the solution to this problem is known, and is the optimal feedback coefficient matrix $K_i^*(t)$. This coefficient, used in the dynamic system equations, defines an optimal covariance matrix and thus defines $\hat{s}_i^*$. In Figure 5.1, $DJ(\hat{s}_i^*)$ is the gradient vector of the functional J at the point $\hat{s}_i^* \in \sigma$.

The theoretical motivation behind this algorithm is the requirement that the necessary conditions for equivalence, given in (A-11), be satisfied. The algorithm tries to bring this condition about by "brute force", by letting $\hat{q}_{i+1} = (1-\gamma)\hat{q}_i + \gamma DJ(\hat{s}_i^*)$, where $\gamma \in [0,1]$ is chosen during each iteration on the basis of engineering judgement. A sketch of DGIM as interpreted in σ-space is given in Figure 5.2. The iteration sequence begins with an arbitrary vector, $\hat{q}_o$, and continues as discussed previously.

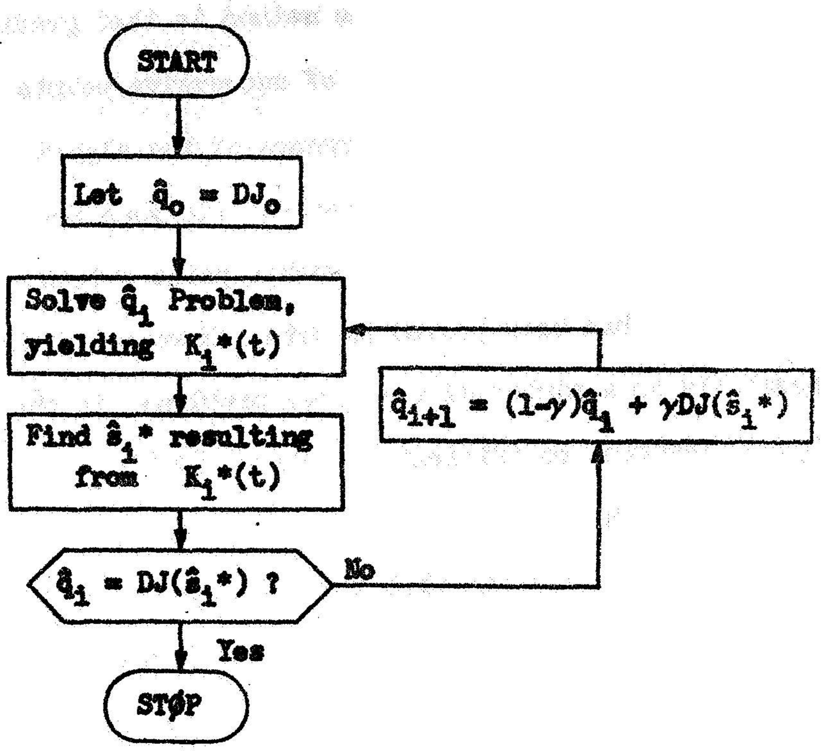


Figure 5.1 Direct Gradient Iteration Method

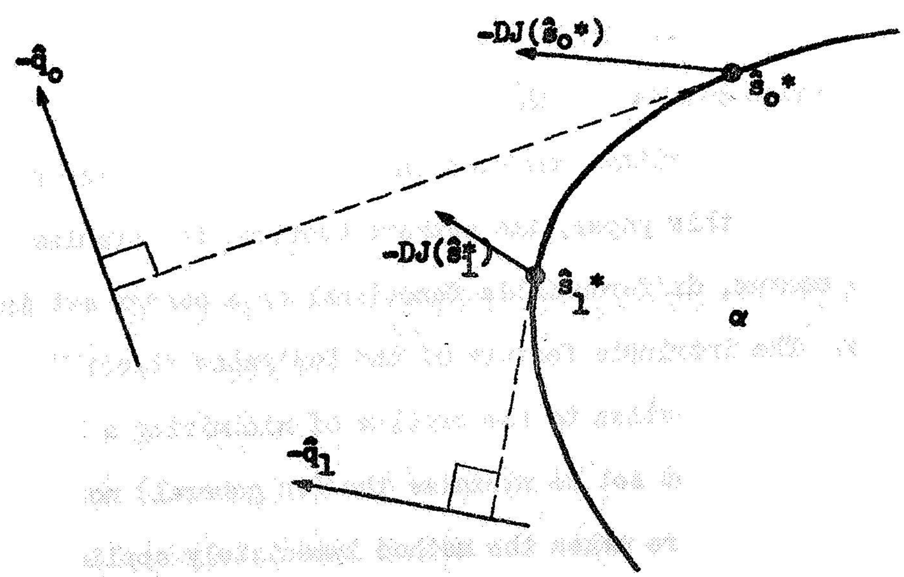


Figure 5.2 DGIM in σ-space

The essential feature of the method is that gradient vectors are the quantities iterated, instead of successive points in α, as is the usual case. So a proof of convergence of the algorithm must show that the sequence of vectors $\hat{q}_i$ "approaches" (in some sense) the gradient of J at its minimum point, assuming such a point exists. No such convergence property has been proved to date. However, Skelton has used DGIM successfully on a number of practical problems, in the sense that it led to "good" feedback coefficients K(t) (see [2.4]). Also, the algorithm was clearly shown to converge in the first example described in Chapter 6. These results indicate that the algorithm is a useful one in certain cases. It is a simple one, and is computationally rapid compared to the perturbed gradient method discussed in Section 5.3. However, it seems to require great care in its use due to its inherent unpredictability.

5.3 Perturbed Gradient Method (PGM)

5.3.1 Description of the Method

The perturbed gradient method described in this section is an application of an algorithm developed in a paper by Dem'yanov and Rubinov [4.1]. In this paper, the authors consider the problem of minimizing a convex, differentiable functional on a convex set in a Banach space. The intrinsic feature of the Dem'yanov algorithm is that it uses the (known) solution to the problem of minimizing a linear functional on the constraint set to minimize the (in general) nonlinear functional. This feature makes the method immediately applicable to the problem of minimizing $J(\hat{s})$ on the constraint set α, since the solution to the problem of minimizing J_Q on α is known.

The name "perturbed gradient method" was taken from an earlier paper by Dem'yanov [5.3], in which PGM and other algorithms were described in Euclidean space. These other algorithms could conceivably be applied to the more general case, but since they are much more complicated than PGM, their usefulness in practice may be restricted.

The PGM algorithm is general enough to include the case in which the minimum of the functional occurs in the interior of the constraint set; however, in the discussion below, it is assumed that the minimum occurs in α_Q (as mentioned in the Introduction to this chapter). The complete set of assumptions under which this algorithm is to be used will be listed in the convergence theorem, Theorem 5.1. These assumptions will be discussed when PGM is applied to the examples in Chapter 6.

A block diagram describing PGM is given in Figure 5.3. The notation used is the same as that in Section 5.2. As can be seen, the stopping condition is the same as that used in DGIM; namely, that the gradient vector of J at the ith solution point be equal to the vector defining the ith J_Q-problem. That is, the equivalence theorem (Theorem 4.1) is again invoked. In practice, of course, it is difficult to make the two vectors equal; however, the norm of the distance between the two can be made as small as desired, within the limits of computational accuracy. This problem of the stopping condition will be discussed more fully in Chapter 6.

The PGM differs from DGIM in that points in the constraint set α are the quantities iterated, instead of gradient vectors. The geometrical significance of the algorithm can be seen from Figure 5.4. An

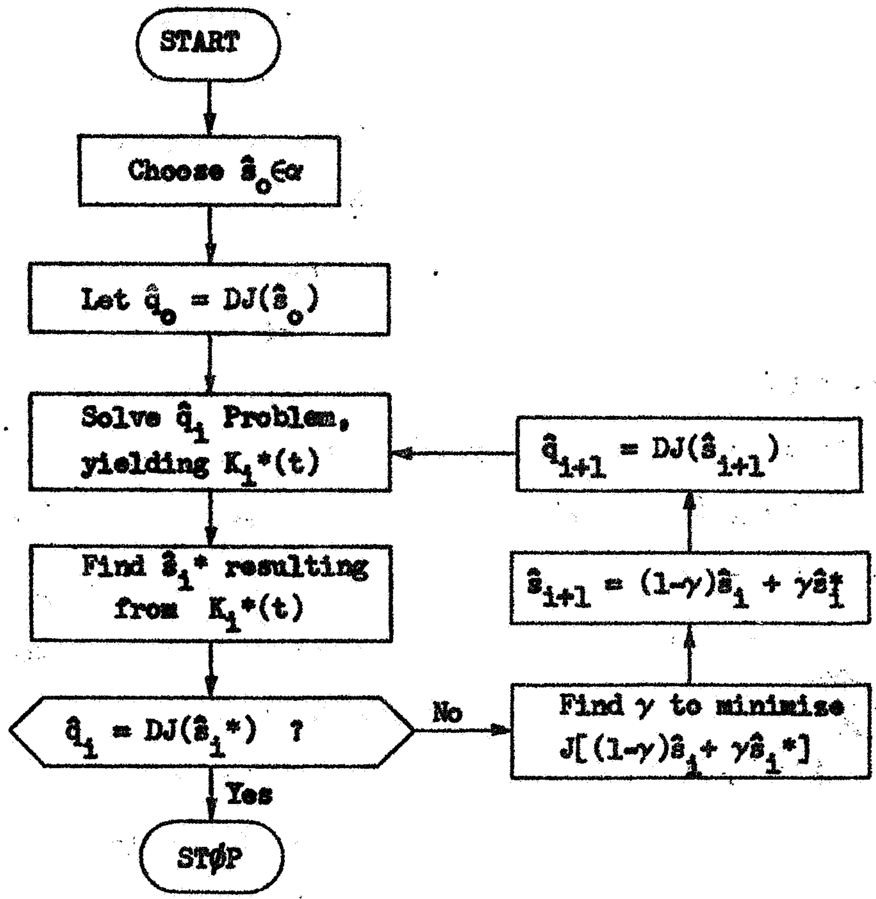
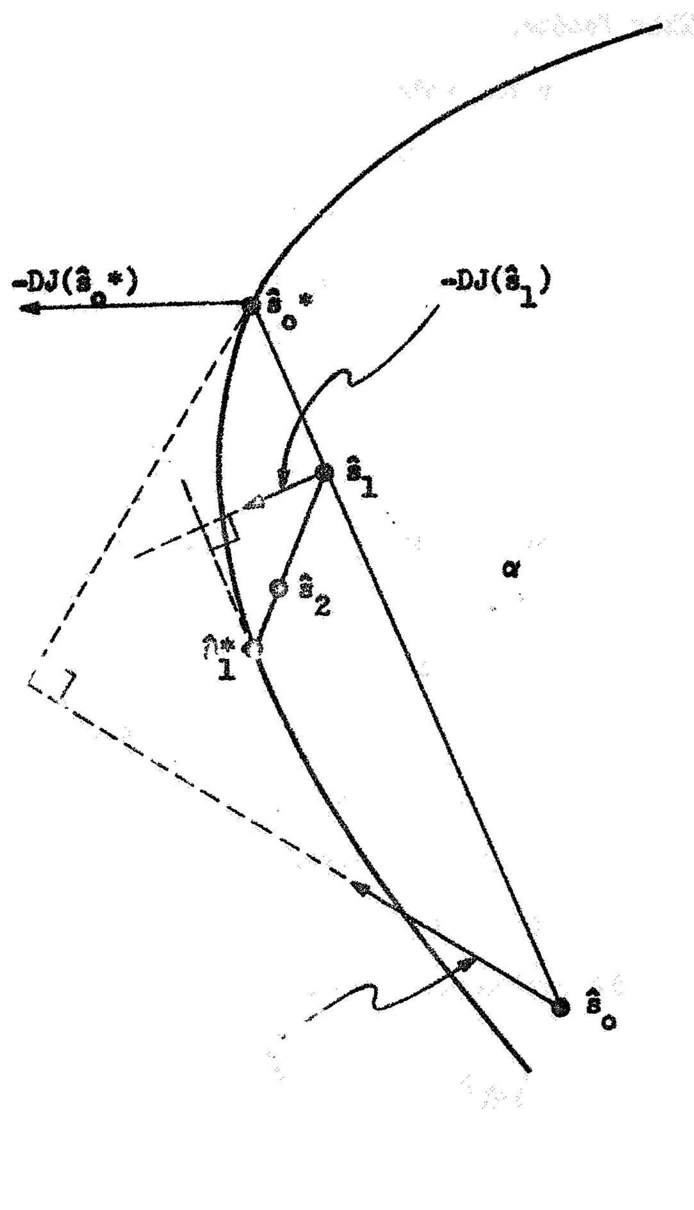


Figure 5.3 Perturbed Gradient Method


-DJ(ŝ$_o$*)
ŝ$_o$*
-DJ(ŝ$_1$)
ŝ$_1$
α
ŝ$_2$
ŝ$_o$

arbitrary point in α is selected as the starting point. This point, $\hat{s}_o$, can be chosen by selecting an arbitrary admissible feedback coefficient, $K_o(t)$. Then $\hat{s}_o$ is defined by the response covariance matrix which results when $K_o(t)$ is used. The gradient vector at $\hat{s}_o$ then defines a "quadratic" problem $\hat{q}_o$, which is solved using the known formulas to yield $K_o^*(t)$. This feedback coefficient $K_o^*(t)$, when used in the system equations, results in the point $\hat{s}_o^* \in \alpha$. Geometrically, solving the $\hat{q}_o$ problem for $\hat{s}_o^*$ corresponds to finding the point in α which is the "farthest" one in the direction of the negative gradient vector. This operation is shown in Figure 5.4 by the orthogonal dotted lines. A "straight line" in σ is then drawn connecting $\hat{s}_o$ and $\hat{s}_o^*$; the next step in the iteration is finding the point on this line at which $J(\hat{s})$ is a minimum. Computationally, this is accomplished by "walking" along the line and sampling values of $J(\hat{s})$ along the way. The assumed convexity of $J(\hat{s})$ assures that the minimum point is unique; so this point can be determined as accurately as required by taking smaller incremented steps along the line. The existence of such a minimum point other than $\hat{s}_o$ itself will be discussed in Section 5.3.2. Once the point is determined, it becomes the next iteration point $\hat{s}_1$, and the iteration is continued by repeating the above procedure. In general, if the ith iteration point is $\hat{s}_i$, the next iteration point is defined by:

$$J(\hat{s}_{i+1}) = \min_{\lambda \in [0,\, 1]} J[(1-\lambda)\hat{s}_i + \lambda\, \hat{s}_i^*] \,, \tag{5-1}$$

where $\hat{s}_i^*$ is the minimum point of the ith J_Q-problem, which is given by $\hat{q}_i = DJ(\hat{s}_i)$. Note that equation (5-1) specifies the new iteration point $\hat{s}_{i+1}$ automatically.

5.3.2 Convergence of the Method

The use of the perturbed gradient method (PGM) described in Section 5.3.1 results in a sequence of points $\{\hat{s}_i\}$, $i = 0,1,2,\ldots$ in α. In this subsection it will be shown that the sequence $\{J(\hat{s}_i)\}$, $i = 0,1,2,\ldots$ converges to $J(\hat{s}^o)$, the minimum value of J on α. The proof of convergence is based in part on a theorem of Dem'yanov and Rubinov in [4.1]. Note that the convergence discussed here is convergence in the performance index J, and not in the sequence of feedback coefficients $\{K_i(t)\}$ or the sequence of points $\{\hat{s}_i\}$, $i = 0,1,2,\ldots$. The results are summarized in Theorem 5.1, which uses the following definition:

<u>Definition 5.1</u>. Let $\hat{s}_o \in \alpha$ be the starting point of the PGM algorithm, and α_Q be defined as in Section 3.5. Then define:

$$\alpha_H = \text{convex hull of } \alpha_Q \cup \hat{s}_o \ ,$$

$$= \left\{ \begin{matrix} \hat{s} : \hat{s} = (1-\lambda)\hat{s}_1 + \lambda\ \hat{s}_2, \text{ for } \hat{s}_1 \text{ and } \hat{s}_2 \\ \text{in } \alpha_Q \cup \hat{s}_o, \text{ and } \lambda \in [0,\ 1] \end{matrix} \right\} \ .$$

The theorem then can be stated:

Theorem 5.1

Assume:

1) the hypotheses of Theorem 4.1 hold;

2) J is a convex functional;

3) α_Q is bounded;

4) $D^2J(\hat{s}, \hat{e})$ (defined in equation (3-22)) is bounded for all $\hat{s} \in \alpha_H$ and all $\hat{e} \in \sigma$ with bounded norm;

5) the perturbed gradient method is defined as in Section 5.3.1, and

generates a sequence of points in α, $\{\hat{s}_i\}$, $i = 0,1,2,\ldots$.

Then:

1) the sequence of values $\{J(\hat{s}_i)\}$, $i = 0,1,2,\ldots$, corresponding to the above $\{\hat{s}_i\}$ sequence, is monotone decreasing;

2) $\lim_{i\to\infty} (DJ(\hat{s}_i), \hat{s}_i^* - \hat{s}_i) = 0$;

3) $\lim_{i\to\infty} J(\hat{s}_i) = J(\hat{s}^o)$; that is, the PGM algorithm converges to a minimum point $\hat{s}^o$ of J.

Proof

1) It must be shown that $J(\hat{s}_{i+1}) < J(\hat{s}_i)$ for an arbitrary $\hat{s}_i$. To prove this, choose $\hat{s}_i$ and let $\hat{q}_i = DJ(\hat{s}_i)$ define the ith quadratic problem. Assuming that $J(\hat{s}_i) \neq J(\hat{s}^o)$, this quadratic problem can be solved, yielding $\hat{s}_i^* \neq \hat{s}_i$. Let

$$\hat{s}_{i\gamma} = \hat{s}_i + \gamma(\hat{s}_i^* - \hat{s}_i), \qquad \gamma \in (0, 1) . \tag{5-2}$$

It will be shown that a γ exists such that $J(\hat{s}_{i\gamma}) < J(\hat{s}_i)$. Hypothesis 1) indicates that the assumptions of Theorem 4.1 are satisfied. Since these assumptions include those of Theorem 3.1, the hypotheses of Lemma 4.1 hold, and the Lemma is valid. Using the finite increment formula in the Lemma, it follows that:

$$J[\hat{s}_i + \gamma(\hat{s}_i^* - \hat{s}_i)] - J(\hat{s}_i) = \gamma(DJ(\hat{s}_i), \hat{s}_i^* - \hat{s}_i) + o(\gamma) . \tag{5-3}$$

Since $\hat{s}_i^*$ is a minimum point of the $\hat{q}_i$-problem, it follows that

$$(DJ(\hat{s}_i), \hat{s}_i^*) \leq (DJ(\hat{s}_i), \hat{s}) \tag{5-4}$$

for all $\hat{s} \in \alpha$. In particular, (5-4) holds for $\hat{s} = \hat{s}_i$. So

$$(DJ(\hat{s}_i),\ \hat{s}_i{}^* - \hat{s}_i) = -M < 0\ , \tag{5-5}$$

where M is some positive real number. Note that the strict inequality holds in (5-5); if it did not, then we would have

$$(DJ(\hat{s}_i),\ \hat{s}_i{}^*) = (DJ(\hat{s}_i),\ \hat{s}_i)\ . \tag{5-6}$$

That is, $\hat{s}_i$ would be a solution of the $\hat{q}_i$-problem defined by $\hat{q}_i = DJ(\hat{s}_i)$. But then by part 2 of Theorem 4.1, $\hat{s}_i$ would be a minimum point of the J-problem, and $J(\hat{s}_i) = J(\hat{s}^o)$. Since it was assumed earlier that $J(\hat{s}_i) \neq J(\hat{s}^o)$, it follows that the strict inequality holds in (5-5).

Using (5-5) in (5-3) results in:

$$J[\hat{s}_i + \gamma(\hat{s}_i{}^* - \hat{s}_i)] - J(\hat{s}_i) = -M\,\gamma + o(\gamma)\ . \tag{5-7}$$

It can be seen that a $\gamma_1 \in (0,\ 1)$ can be found such that the right side of (5-7) becomes negative. For this γ_1, (5-7) implies (using (5-2)) that:

$$J(\hat{s}_{i\gamma_1}) = J[\hat{s}_i + \gamma_1(\hat{s}_i{}^* - \hat{s}_i)] < J(\hat{s}_i)\ . \tag{5-8}$$

Using the definition of $\hat{s}_{i+1}$ in (5-1), equation (5-8) becomes

$$J(\hat{s}_{i+1}) = \min_{\gamma \in (0,1)} J(\hat{s}_{i\gamma}) \leq J(\hat{s}_{i\gamma_1}) < J(\hat{s}_i)\ , \tag{5-9}$$

and part 1) of the Theorem is proved.

2) Since J is bounded below on α (by the assumption in Theorem 4.1 that a minimum of J on α exists), and since the sequence $\{J(\hat{s}_i)\}$ is monotone decreasing by part 1) of the Theorem, the limit

$$\lim_{i \to \infty} J(\hat{s}_i) = L > -\infty \qquad (5\text{-}10)$$

exists.

Equation (5-1) defining PGM can be written

$$J(\hat{s}_{i+1}) = \min_{\gamma \in (0,1)} J[\hat{s}_i^* + \gamma(\hat{s}_i - \hat{s}_i^*)] \;, \qquad (5\text{-}11)$$

from which the following inequality holds for $\gamma \in (0,1)$:

$$\begin{aligned} J(\hat{s}_{i+1}) &\leq J[\hat{s}_i^* + \gamma(\hat{s}_i - \hat{s}_i^*)] \\ &= J[\hat{s}_i + (1-\gamma)(\hat{s}_i^* - \hat{s}_i)] \end{aligned} \qquad (5\text{-}12)$$

Using equation (4-3) from Lemma 4.1, (5-12) becomes:

$$\begin{aligned} J(\hat{s}_{i+1}) &\leq J(\hat{s}_i) + (1-\gamma)(DJ(\hat{s}_i),\, \hat{s}_i^* - \hat{s}_i) \\ &+ \tfrac{1}{2}(1-\gamma)^2(D^2J(\hat{s}_i + \beta(\hat{s}_i^* - \hat{s}_i)), \hat{s}_i^* - \hat{s}_i), \hat{s}_i^* - \hat{s}_i) \;, \end{aligned} \qquad (5\text{-}13)$$

where $\beta \in [0, (1-\gamma)]$.

By the Schwarz inequality,

$$\begin{aligned} &(D^2J(\hat{s}_i + \beta(\hat{s}_i^* - \hat{s}_i)),\, \hat{s}_i^* - \hat{s}_i),\, \hat{s}_i^* - \hat{s}_i) \\ &\leq \|D^2J(\hat{s}_i + \beta(\hat{s}_i^* - \hat{s}_i)), \hat{s}_i^* - \hat{s}_i)\|\sigma \cdot \|\hat{s}_i^* - \hat{s}_i\|\sigma \;, \end{aligned} \qquad (5\text{-}14)$$

where $\|\cdot\|\sigma$ is defined in (3-11).

It will now be shown that the right side of (5-14) is bounded for all i. The point $\hat{s}_i + \beta(\hat{s}_i^* - \hat{s}_i)$ is in α_H by definition 5.1 and the construction of $\hat{s}_i$ using the PGM algorithm. So, using hypothesis 4), the right side of (5-14) is bounded if $\|\hat{s}_i^* - \hat{s}_i\|\sigma$ is bounded. We have:

$$\|\hat{s}_i^* - \hat{s}_i\|\sigma \leq \|\hat{s}_i^*\|\sigma + \|\hat{s}_i\|\sigma \quad . \tag{5-15}$$

Since $\hat{s}_i^*$ and $\hat{s}_i$ are both in α_H for all i, it is then sufficient to show that α_H is bounded. If $\hat{s} \in \alpha_H$, then (using Definition 5.1 and the triangle inequality):

$$\|\hat{s}\|\sigma \leq \|\hat{s}_1\|\sigma + \|\hat{s}_2\|\sigma \quad , \tag{5-16}$$

where $\hat{s}_1$ and $\hat{s}_2$ are in $\hat{s}_o \cup \alpha_Q$. But $\hat{s}_o$ is a single point and α_Q is bounded (by hypothesis 3); so α_H is bounded, and (5-14) can then be rewritten:

$$(D^2J(\hat{s}_i + \beta(\hat{s}_i^* - \hat{s}_i), \hat{s}_i^* - \hat{s}_i), \hat{s}_i^* - \hat{s}_i) \leq N < \infty \tag{5-17}$$

for some positive real N. Then (5-13) can be rewritten:

$$\begin{aligned} J(\hat{s}_{i+1}) &\leq J(\hat{s}_i) + (1-\gamma)(DJ(\hat{s}_i), \hat{s}_i^* - \hat{s}_i) + \tfrac{1}{2}(1-\gamma)^2 N \\ &= J(\hat{s}_i) + (1-\gamma)[(DJ(\hat{s}_i), \hat{s}_i^* - \hat{s}_i) + \tfrac{1}{2}(1-\gamma)N], \end{aligned} \tag{5-18}$$

for $\gamma \in [0,1)$.

Note that, by definition of the element $\hat{s}_i^*$, we have:

$$(DJ(\hat{s}_i),\ \hat{s}_i^* - \hat{s}_i) \leq 0,\ i = 1,2,\ldots \quad . \tag{5-19}$$

Suppose now that part 2) of the Theorem were false. Then a sequence $\hat{s}_{i_k}$ and a $\rho > 0$ can be found, such that

$$(DJ(\hat{s}_i),\ \hat{s}_i^* - \hat{s}_i) \leq -\rho < 0,\ k = 1,2,\ldots \quad . \tag{5-20}$$

In this case, (5-18) becomes:

$$J(\hat{s}_{i_{k+1}}) \leq J(\hat{s}_{i_k}) + (1-\gamma)[-\rho + \tfrac{1}{2}(1-\gamma)N] \quad . \tag{5-21}$$

Passing to the limit as $k \to \infty$, we have:

$$L \leq L + (1 - \gamma)[- \rho + \tfrac{1}{2}(1 - \gamma)N] \; ; \tag{5-22}$$

or, since $(1 - \gamma) > 0$,

$$- \rho + \tfrac{1}{2}(1 - \gamma)N \geq 0 \; . \tag{5-23}$$

But (5-23) does not hold if γ is chosen such that $(1 - \gamma) < 2\rho/N$. So a contradiction results, and the second part of the Theorem is proved.

3) By equation (4-3) of Lemma 4.1, for any $\hat{s} \in \alpha$ we can write:

$$\begin{aligned} J(\hat{s}) - J(\hat{s}_i) &= (DJ(\hat{s}_i), \hat{s} - \hat{s}_i) \\ &\quad + \tfrac{1}{2}(D^2J(\hat{s}_i + \beta(\hat{s} - \hat{s}_i), \hat{s} - \hat{s}_i), (\hat{s} - \hat{s}_i)) \; . \end{aligned} \tag{5-24}$$

Since J is convex, the second term on the right side of (5-24) is nonnegative; so

$$J(\hat{s}) - J(\hat{s}_i) \geq (DJ(\hat{s}_i), \hat{s} - \hat{s}_i) \quad . \tag{5-25}$$

Taking the minimum (in $\hat{s}$) of both sides of (5-25) on α, and remembering that J is minimized at $\hat{s}^o$, we have:

$$\begin{aligned} J(\hat{s}^o) - J(\hat{s}_i) &\geq \min_{\hat{s} \in \alpha} (DJ(\hat{s}_i), \hat{s} - \hat{s}_i) \\ &= (DJ(\hat{s}_i), \hat{s}_i{}^* - \hat{s}_i) \; . \end{aligned} \tag{5-26}$$

So

$$(DJ(\hat{s}_i), \hat{s}_i - \hat{s}_i{}^*) \geq J(\hat{s}_i) - J(\hat{s}^o) \geq 0 \; . \tag{5-27}$$

From part 2) of the Theorem, the left side of (5-27) goes to zero as $i \to \infty$. So part 3) follows from (5-27). Q.E.D.

The above Theorem is useful in that it guarantees convergence of the PGM algorithm if the hypotheses are satisfied. As mentioned before, no such theorem is presently available for the DGIM algorithm. In that algorithm, the sequence $\{J(\hat{s}_i)\}$, $i = 0,1,\ldots$ is not even monotone decreasing (in general). The computer results described in Chapter 6 verify the monotonicity of the $J(\hat{s}_i)$ sequence when PGM is used, while the DGIM results are more erratic.

Theorem 5.1 is an additional demonstration that the function space formulation is a useful one. The formulation led to the development of the PGM, and also allows the function space results of Dem'yanov in [4.1] to be applied to the above convergence proof.

5.3.3 Comments on the Method

The PGM algorithm has certain points of similarity with the iterative procedure of Gilbert [5.4]. In the case in which $J(\hat{s})$ is a quadratic form in $s(T)$ and $s(t)$, the two methods are identical (except that Gilbert's method is formulated in Euclidean space instead of Hilbert space). Neither method requires an explicit expression for the constraint set; all that is required is the availability of a method for solving "linear programs" (Gilbert's term) on the constraint set. This solution of a linear program is Gilbert's "contact function", and corresponds to solving the J_Q-problem in PGM.

Computationally, the main problem in PGM is finding the minimum point along the "line" connecting $\hat{s}_i$ and $\hat{s}_i{}^*$. The repeated evaluation of $J(\hat{s})$ involves computation of an integral (see equation (3-16)), and may be time-consuming. Some methods for decreasing the time and storage required to evaluate $J(\hat{s})$ are described in Appendix G. The results in

Chapter 6 show that PGM takes at least twice the computer time of DGIM per iteration. However, PGM is more dependable, since the successive values of the performance index are always decreasing. This makes PGM more efficient in terms of speed of performance index minimization. Also, PGM is sure to converge if the conditions of Theorem 5.1 are met; no such assurance is available for DGIM.

Note that the quantity of interest in the solution of the J-problem is the optimal feedback coefficient $K^*(t)$. Both the PGM and DGIM algorithms give a nonoptimal feedback coefficient $K_i(t)$ at each iteration step. This feature is important in engineering applications, since a truly optimal coefficient may not be of interest. In this case, the iteration will only be continued until $K_i(t)$ gives "acceptable" system performance when used in the system equations.

CHAPTER 6

COMPUTATIONAL RESULTS

6.1 Introduction

The PGM and DGIM algorithms discussed in Chapter 5 were applied to two stochastic control problems, and the results are summarized in this chapter. The first problem considered is that of controlling a pure inertia, which is disturbed by filtered white noise. The performance index in this example is the square of the norm of $\hat{s}\epsilon\sigma$, where $\hat{s}$ (defined in (3-15)) represents the system response covariances. The second problem considered is that of reducing wind-gust effects on a large missile during the boost phase of flight. The performance index used in this example is one derived by Skelton in [6.1], and is an upper bound on the probability that certain system responses will exceed their given bounds. Because of computational difficulties with Skelton's performance index, a new performance index that "matches" Skelton's in a certain sense is introduced. The PGM algorithm is then applied to the problem of minimizing this index to get good load-relief controllers for the launch booster.

The algorithms were programmed in Fortran IV to run on the IBM 7094 (first example) and the CDC 6500 (second example) computing systems. A description of the programs used and some computational techniques are given in Appendix G.

The computational results indicate that PGM is a more dependable algorithm than DGIM, since the sequence of values of the performance index that it generates is monotone decreasing. In the first example, however, PGM takes about twice as much computer time to run (per iteration) as does DGIM. In the second example, the two algorithms take about the same amount of time. The successful use of PGM in the second example shows that it is applicable to high-order systems in practical problems. The second example also displays a "suboptimal" approach to Skelton's load-relief problem, and indicates that useful controls can be generated by PGM and the supporting function-space theory.

6.2 A Minimum Norm Problem

6.2.1 Problem Statement

The stochastic system to be considered in this section is essentially a pure inertia (or double integrator) disturbed by filtered white noise. A block diagram of the system is shown in Figure 6.1. The filter input, n_1, and the measurement noise, w_1 and w_2, are white Gaussian noise. The system output, θ, can be considered an angular displacement, $\dot{\theta}$ angular rate, and I the moment of inertia.

To put the system equations in the form given in Chapter 2, identify the vectors x, v, and r as follows (the subscripts indicate vector components):

$$\begin{array}{lll} x_1 = \theta & v_1 = 0 & r_1 = x_1 \\ x_2 = \dot{\theta} & v_2 = 0 & r_2 = x_2 \\ x_3 = n_2 & v_3 = n_1 & r_3 = u \end{array} \tag{6-1}$$

The measurement vector z is given by:

$$\begin{aligned} z_1 &= x_1 + w_1 \\ z_2 &= x_2 + w_2 \end{aligned} \tag{6-2}$$

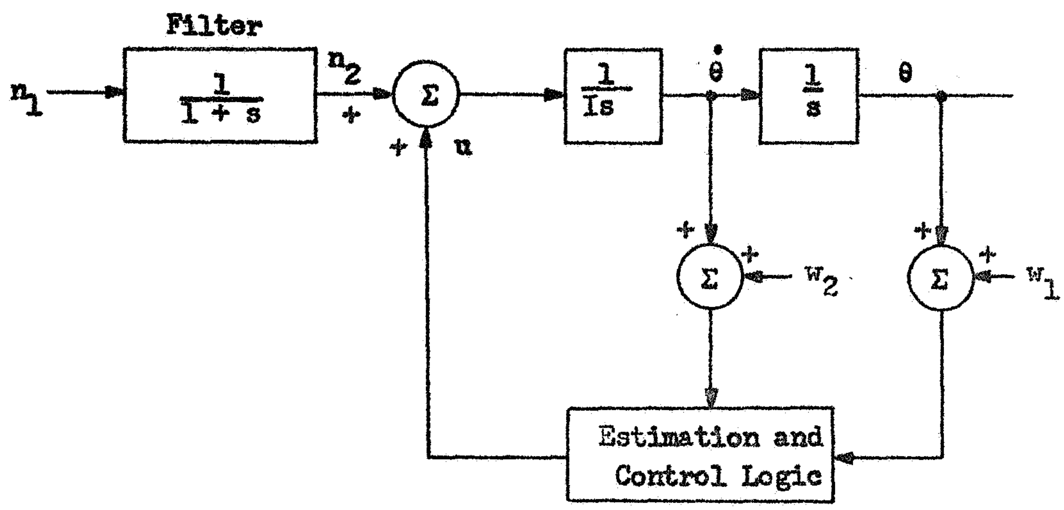


Figure 6.1 Pure Inertia System

The noise vector w has components w_1 and w_2, as shown in Figure 6.1. Using the above identifications, the dynamic system equations are:

$$\dot{x}_1 = x_2 \tag{6-3}$$

$$\dot{x}_2 = (x_3 + u)/I \tag{6-4}$$

$$\dot{x}_3 = - x_3 + v_3 \ . \tag{6-5}$$

Note that the control u is a scalar, and the state x_3 is the output of the noise filter. For definiteness, the following parameters will be used:

$$I = 100$$

$$E[v_3^2(t)] = 0.01$$

$$E[w_i^2(t)] = 0.1, \quad i = 1,2,3.$$

$$t_o = 0$$

$$T = 10.$$

Using these numbers, the parameter matrices are as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0.01 \\ 0 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.01 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and the noise covariance matrices are specified by

$$N_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$N_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

The performance index to be minimized is the square of the norm of the element $\hat{s}$ in the space σ. That is,

$$\begin{aligned} J(\hat{s}) &= \tfrac{1}{2}(\hat{s}, \hat{s}) \\ &= \tfrac{1}{2}s(T)\cdot s(T) + \tfrac{1}{2}\int_{t_o}^{T} s(t)\cdot s(t)dt . \end{aligned} \tag{6-6}$$

Minimizing this index can be interpreted as reducing the effect of the disturbance noise on θ and $\dot{\theta}$, while putting a penalty on the control u.

The problem to be solved is then the general problem discussed in Section 2.2, using the parameter matrices given above and the performance index in (6-6). It is to find the $u \in U$ (defined in (2-9)) such that $J(\hat{s})$ in (6-6) is minimized, subject to the system equations (2-1) to (2-8). In geometrical terms in function space, the problem is simply to find the feedback coefficient $K^*(t)$ that produces $\hat{s}^*$, the element of minimum norm in α (the set of attainability).

Since the PGM algorithm will be applied to the above example, a few comments will be made concerning the hypotheses in the theorems derived in Chapters 3, 4, and 5. Theorem 3.1 requires that $J(\hat{s})$ be defined for every $\hat{s}$ in σ; this is certainly the case for the J in (6-6). Comparing the definition of $J(\hat{s})$ in (3-16) with the specific J in (6-6) yields the following expressions for f_1 and f_2:

$$f_1[s(T)] = \tfrac{1}{2}s(T)\cdot s(T), \quad f_2[s(t)] = \tfrac{1}{2}s(t)\cdot s(t) \; . \qquad (6\text{-}7)$$

Using the definitions of $\frac{\partial f_i}{\partial s}$ and $\frac{\partial^2 f_i}{\partial s^2}$ in (3-26) results in:

$$\frac{\partial f_1(T)}{\partial s} = s(T) \; , \quad \frac{\partial f_2(t)}{\partial s} = s(t) \; ; \qquad (6\text{-}8)$$

$$\frac{\partial^2 f_i}{\partial s^2} = I \; , \quad i = 1,2, \qquad (6\text{-}9)$$

where I is the $(\ell^2 \times \ell^2)$ identity matrix.

By (6-7), (6-8), and (6-9) it is seen that hypothesis 2) of Theorem 3.1 is satisfied. From (3-20) and (3-22), we have:

$$DJ(\hat{s}) = \hat{s} \; , \qquad (6\text{-}10)$$

$$D^2 J(\hat{s}, \hat{e}) = \hat{s} \; . \qquad (6\text{-}11)$$

The quantities in (6-10) and (6-11) are certainly continuous in $\hat{s}$ in the norm of the σ-space; so the hypothesis in Part 2 of Theorem 3.1 is also satisfied, and therefore the theorem can be applied to the minimum norm example. Theorem 3.2 has no hypotheses under question, since it is merely on assertion concerning the solution of the J_Q-problem. Lemma 4.1 has the same hypotheses as Theorem 3.1, so it is applicable to the example.

In the hypotheses of Theorem 4.1, the convexity of α and the existence of a minimum point of the J-problem cannot be verified at present. By (6-8) and the "stacking" procedure used to form $\hat{s}$, the matrices $\frac{\partial f_1}{\partial S}$ and $\frac{\partial f_2}{\partial S}(t)$ are:

$$\frac{\partial f_1}{\partial S} = S(T), \quad \frac{\partial f_2}{\partial S}(t) = S(t); \tag{6-12}$$

The matrices $S(T)$ and $S(t)$ are covariance matrices and thus are always positive semidefinite (see part 2 of Theorem 4.2); this proves the first part of hypothesis 3) in Theorem 4.1. The second part cannot be presently verified, but the fourth hypothesis is valid from previous discussion. The assumption in part 2) of Theorem 4.1 concerns the convexity of J; since the J in the minimum norm problem is quadratic in the norm of $\hat{s}$, it can be easily shown that it is a convex functional. In Theorem 5.1, hypotheses 1) and 2) have been discussed previously, and 4) follows easily from equation (6-11). The boundedness of α_Q cannot be presently verified, however.

In general, the minimum norm example satisfies most of the hypotheses in the theorems derived. The most serious exceptions are, of course, the assumptions concerning the convexity of α and the existence of a solution to the J-problem. The success obtained in using the PGM and DGIM algorithms suggests, however, that the above theorems are valid for this example.

6.2.2 Results and Discussion

The above problem was solved using both the DGIM and the PGM algorithms described in Chapter 5. As mentioned in that chapter, the goal of the algorithms was to find a J_Q-problem that was equivalent to the above J-problem. In this section, DGIM and PGM are compared, and the results of the computational solutions are given and discussed.

Some of the computer techniques used and a description of the programs which implement the algorithms are given in Appendix G.

Two variables which were not defined in Chapter 5 will be used in the evaluation of the computational results. These variables will now be defined. Using the notation of Chapter 5, let

$$\Delta_i = \left\| \frac{\hat{q}_i}{\|\hat{q}_i\|_\sigma} - \frac{DJ(\hat{s}_i*)}{\|DJ(\hat{s}_i*)\|_\sigma} \right\|_\sigma . \tag{6-13}$$

To get a geometric interpretation of Δ_i in σ-space, refer to Figure 5.2 or 5.4. Since Δ_i is the norm of the difference of two unit vectors, it is a measure of the "angle" between these two vectors. Thus Δ_i is a measure of how well the necessary conditions for equivalence, given in part 1) of Theorem 4.1, are being satisfied at the ith iteration. Suppose also that an $\hat{s}_i*$ is found such that $\Delta_i = 0$ for that i. Then, by (6-13), $\hat{s}_i*$ is the solution of the J_Q-problem defined by $\hat{q}_i = DJ(\hat{s}_i*)$. This means that $\hat{s}_i*$ satisfies the conditions in part 2) of Theorem 4.1, and is thus a minimum point of the J-problem (the desired solution point). From the above reasoning, the size of Δ_i is a good measure of how well $\hat{s}_i*$ satisfies the equivalence conditions. Thus $\Delta_i \leq \delta$ for some small real δ was chosen as a stopping condition for both the PGM and the DGIM algorithms. Also, the $\hat{s}_i*$ which resulted when the stopping condition was met was regarded as satisfying the conditions of Part 2) in Theorem 4.1 "approximately". This point was then considered an "approximate" minimum point of J.

Suppose now that the above stopping condition has been satisfied, and let $\hat{s}^o$ be the point in α at which the sufficient conditions for equivalence in Part 2) of Theorem 4.1 have been satisfied "approximately". Then the following quantity is defined:

$$\Delta_i^{\,o} = \left\| \frac{\hat{q}_i}{\|\hat{q}_i\|_\sigma} - \frac{DJ(\hat{s}^o)}{\|DJ(\hat{s}^o)\|_\sigma} \right\|_\sigma . \tag{6-14}$$

If $J(\hat{s})$ has several minimum points in α, $\Delta_i^{\,o}$ may not converge to zero as $i \to \infty$. If it does, however, it can be used as another measure (in addition to Δ_i) of the quality of convergence of the algorithm considered.

The specific DGIM algorithm which was used to solve the minimum norm problem is given in Figure 6.2. In the description of DGIM in Chapter 5, the coefficient γ was left to be an arbitrary number between 0 and 1. In the actual algorithm used, γ was initially 0.9; that is, since the initial guess of $\hat{q}_o$ was probably a poor one, the gradient vector of J at $\hat{s}_i{}^*$ was weighted heavily in the expression for $\hat{q}_{i+1}$. If Δ_i started to increase (the algorithm began to diverge), γ was reduced by 0.1 to stabilize the iteration procedure. It will be seen that this method of choosing γ worked well for the problem considered. As mentioned before, the stopping condition for DGIM was linked to Δ_i; the iteration was terminated when Δ_i became less than 0.01. The number 0.01 was chosen arbitrarily; however, its use resulted in a good compromise between the requirement that the necessary conditions be satisfied and the requirement that computer time and accuracy not be excessive.

The PGM algorithm used is given in Figure 6.3. The initial guess of $\hat{q}_o$ was made by choosing an arbitrary feedback coefficient, $K_o(t)$, computing the resulting point $\hat{s}_o \in \alpha$, and letting $\hat{q}_o$ equal the gradient vector of J at $\hat{s}_o$. The same stopping condition as described above was also used in the PGM algorithm.

The computational results using the two algorithms are shown in

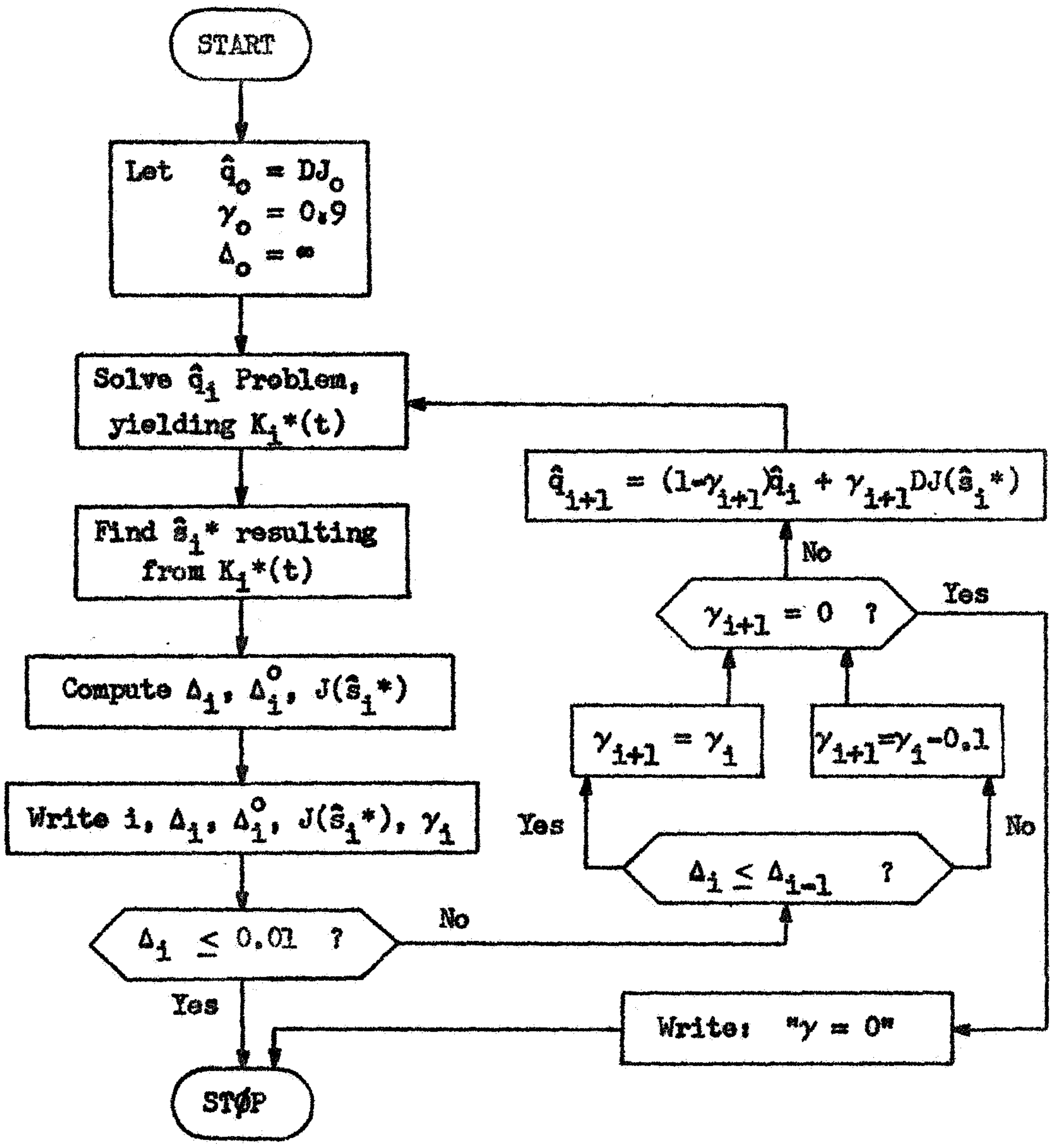


Figure 6.2 DGIM Algorithm - Minimum Norm Problem

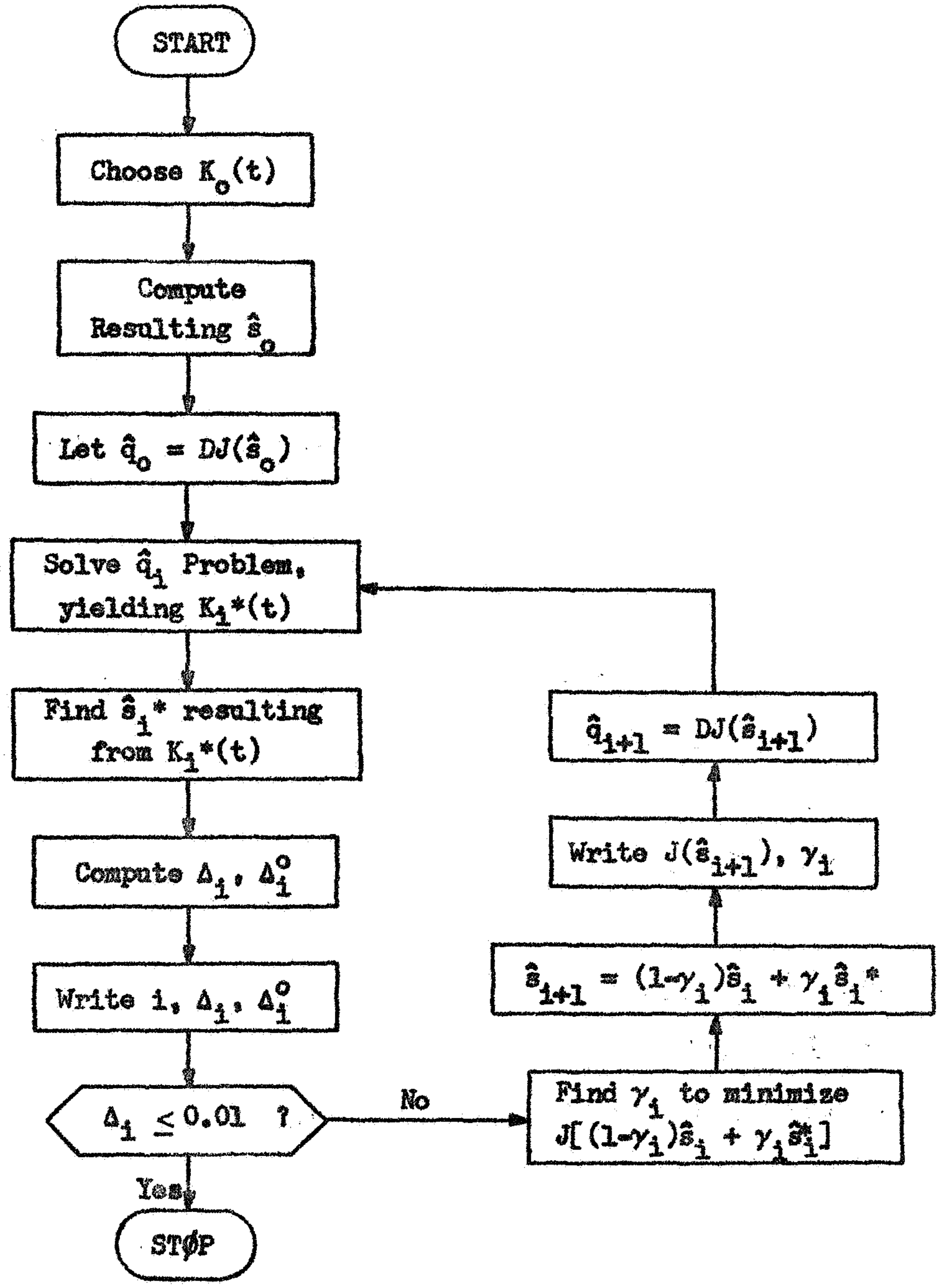


Figure 6.3 PGM Algorithm - Minimum Norm Problem

Figures 6.4, 6.5, and 6.6. The results are plotted as a function of computer time, so that the methods can be compared on the same basis. The PGM method took about twice as long to run, per iteration, as did DGIM; so a comparison of convergence on the basis of number of iterations would not be meaningful. (Each point on the graphs represents an iteration).

It can be seen from the figures that both algorithms achieved the stopping condition ($\Delta_i \leq 0.01$), but that the nature of convergence is different for each method. (Note that a few iterations were made after the stopping condition was satisfied). The successive values of $J(\hat{s}_i)$ are monotonically decreasing for PGM, as would be expected from the nature of the algorithm (see Theorem 5.1). Also, Δ_i° decreased monotonically. This makes sense, by the following reasoning. For this problem, $DJ(\hat{s}) = \hat{s}$. Then, by definition, when $J(\hat{s}_i) \downarrow J(\hat{s}_o)$ (converges by monotonically decreasing to $J(\hat{s}_o)$), we have that $\|\hat{s}_i - \hat{s}_o\|\sigma \to 0$; and so $\|DJ(\hat{s}_i) - DJ(\hat{s}_o)\|\sigma \to 0$. By (6-14) and the fact that $\hat{q}_i = \nabla J(\hat{s}_i)$, it follows that $\Delta_i^{\circ} \downarrow 0$ as $i \to \infty$. It should be noted that the $\hat{s}^{\circ}$ used to compute Δ_i° was the point obtained computationally by PGM and DGIM when the stopping condition was satisfied. So this $\hat{s}^{\circ}$ did not satisfy the equivalence conditions exactly, but only within the tolerance specified by the stopping condition. The behavior of Δ_i for PGM, as shown in Figure 6.5, is considerably more erratic than that of $J(\hat{s}_i)$ and Δ_i°. This behavior is possible because PGM chooses values of $\hat{s}_i$ to decrease $J(\hat{s}_i)$; it does not matter what the gradient vector of J at these points happens to be. So in the example considered, the point chosen at (computer time) 7.5 minutes resulted in a decrease in J, but the gradient vector at this point, $DJ(\hat{s}_i)$, did not compare very well with the gradient

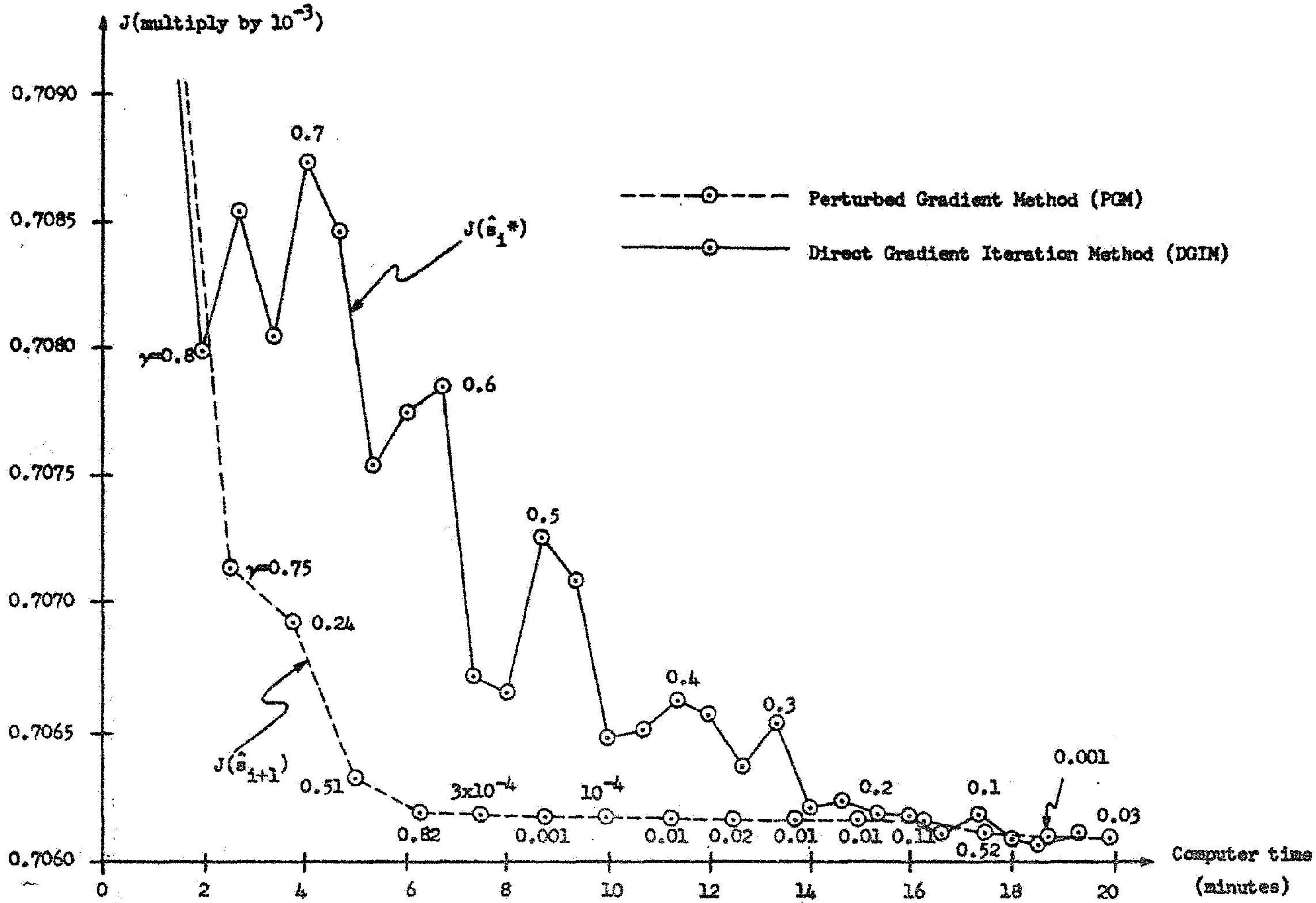


Figure 6.4 Performance Index - Minimum Norm Problem

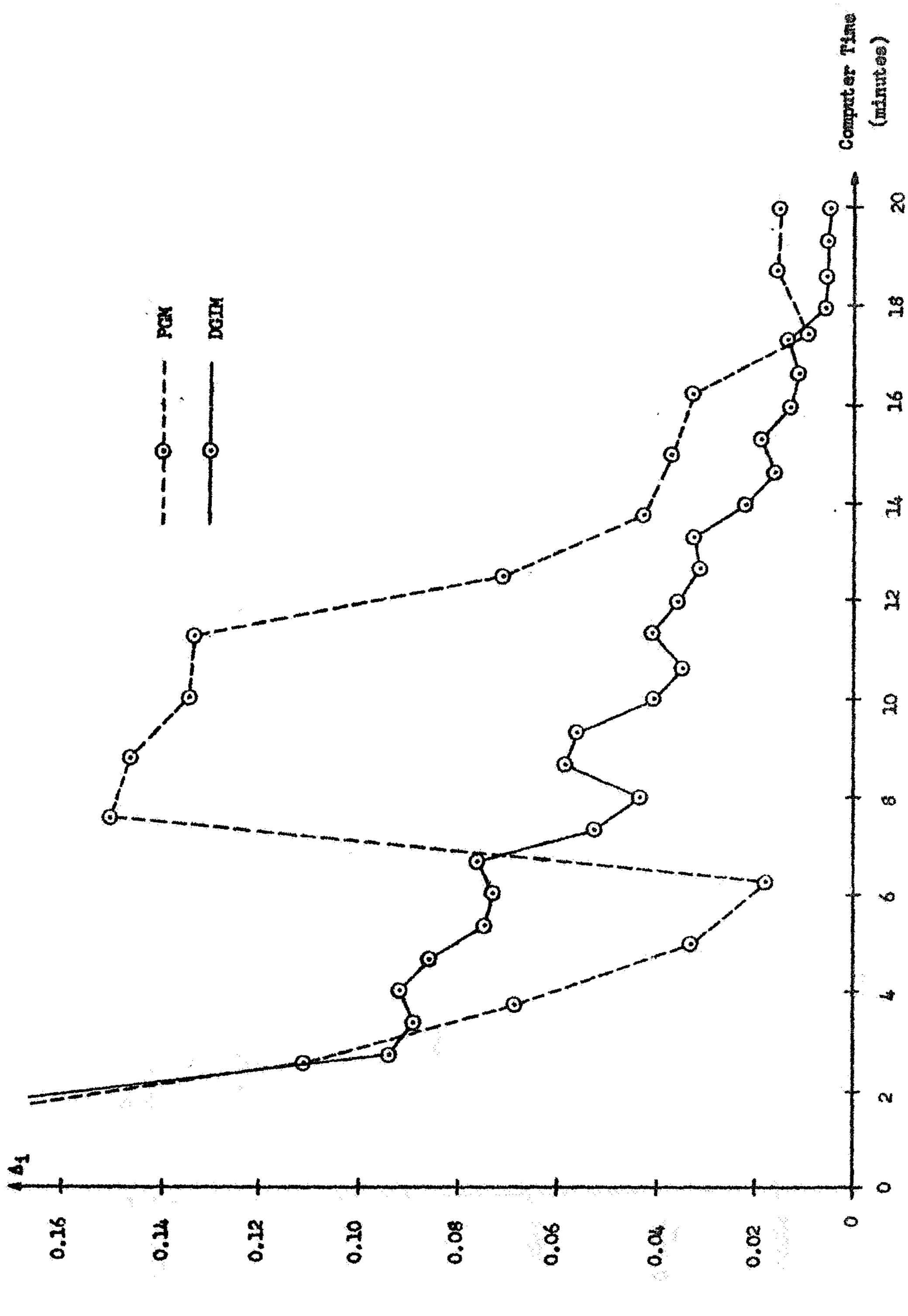


Figure 6.5 Δ_1 for Minimum Norm Problem

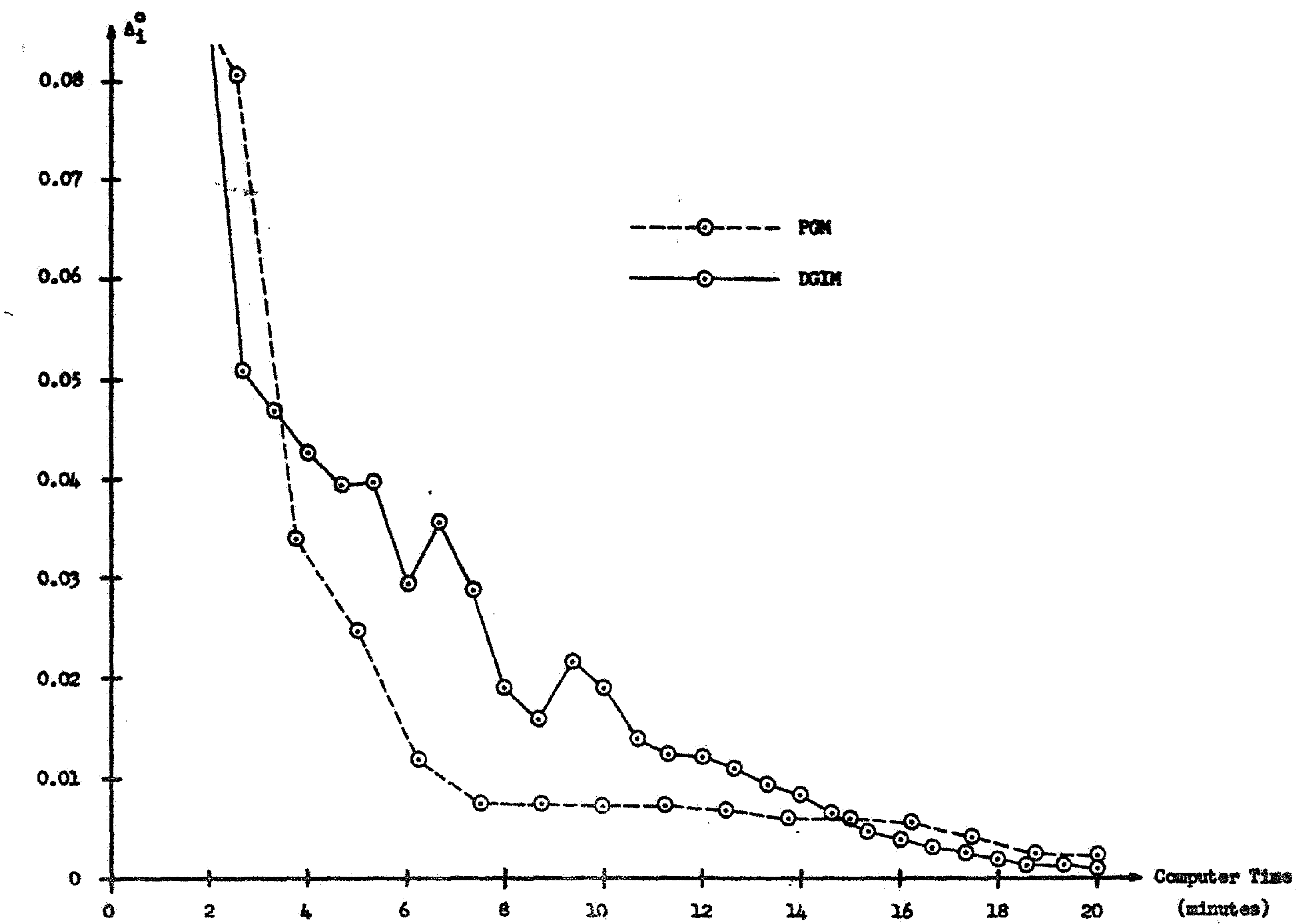


Figure 6.6 Δ_i^o for Minimum Norm Problem

vector at the resulting solution point, $DJ(\hat{s}_i^*)$. The behavior of PGM as discussed above is consistent with the inherent nature of the algorithm.

The nature of DGIM is also reflected in the results shown in Figures 6.4, 6.5, and 6.6. Remember that the choice of γ_i was made by checking the convergence of the method as reflected by Δ_i. If Δ_i began to increase, γ_i was reduced by 0.1, which would hopefully stabilize the algorithm and cause Δ_i to decrease again. As shown in Figure 6.5, this is what actually occurred. Thus the stopping condition was eventually achieved using the above method. The interesting result is that, by "forcing" Δ_i to become small, the algorithm also causes $J(\hat{s}_i^*)$ to become small, as shown in Figure 6.4. In a way, this is an experimental verification of the sufficiency of the equivalence condition in Part 2) of Theorem 4.1 for the example considered. Similarly, the reduction of Δ_i by reducing $J(\hat{s}_i)$ in the PGM algorithm can be viewed as verifying the necessity of the equivalence condition.

In general, PGM seems to be a more dependable algorithm than DGIM, because $J(\hat{s}_i)$ decreases monotonically. Also, a convergence theorem (Theorem 5.1) exists for PGM, while no such theorem exists for DGIM. In a practical application, PGM probably would have been stopped after five iterations, because the increase in system performance, as reflected in the value of J, was relatively small after that. The same can be said if the stopping condition was chosen to be $\Delta_i \leq 0.02$ instead of $\Delta_i \leq 0.01$.

The initial conditions for the results described above were found by using $K_o(t) = [1\ 1\ 1]$ as the initial feedback coefficient. This coefficient was the initial condition for PGM. When $K_o(t)$ was used in the system equations, it resulted in the point $\hat{s}_o$. Then $\hat{q}_o = DJ(\hat{s}_o)$ was used as the initial condition for DGIM, thus assuring that the two algorithms were started on an equal basis. The components of the feedback coefficient $K^o(t)$, which was computed when the stopping condition $\Delta_i \leq 0.01$ was satisfied, are plotted in Figure 6.7. This feedback coefficient defines the optimal control for the minimum norm problem (by (2-9)), within the accuracy of the stopping condition. The diagonal elements of the response covariance matrix $S^o(t)$, which resulted when $K^o(t)$ was used in the system equations, are plotted in Figure 6.8 as a function of time.

It should be mentioned that a set of comparison runs using an initial feedback coefficient $K_o(t) = [10\ 10\ 10]$ was also made, and that both the PGM and DGIM algorithms converged to the same solution ($\hat{s}^o$) as found above. The nature of the convergence was similar to that shown in Figures 6.4, 6.5, and 6.6, so those results are not given here.

6.3 Skelton's Launch Booster Gust Alleviation Problem

6.3.1 Problem Statement

The problem considered in this section concerns the alleviation of wind-gust effects on launch boosters, and was formulated and studied by Skelton in [6.1]. As was stated in Chapter 1, this wind-gust problem motivated the research recorded in this thesis; therefore, it is natural to use the results of the research to attack the original problem. The equations which model the booster-pitch-axis dynamics and the filter describing the incident winds were derived in detail in

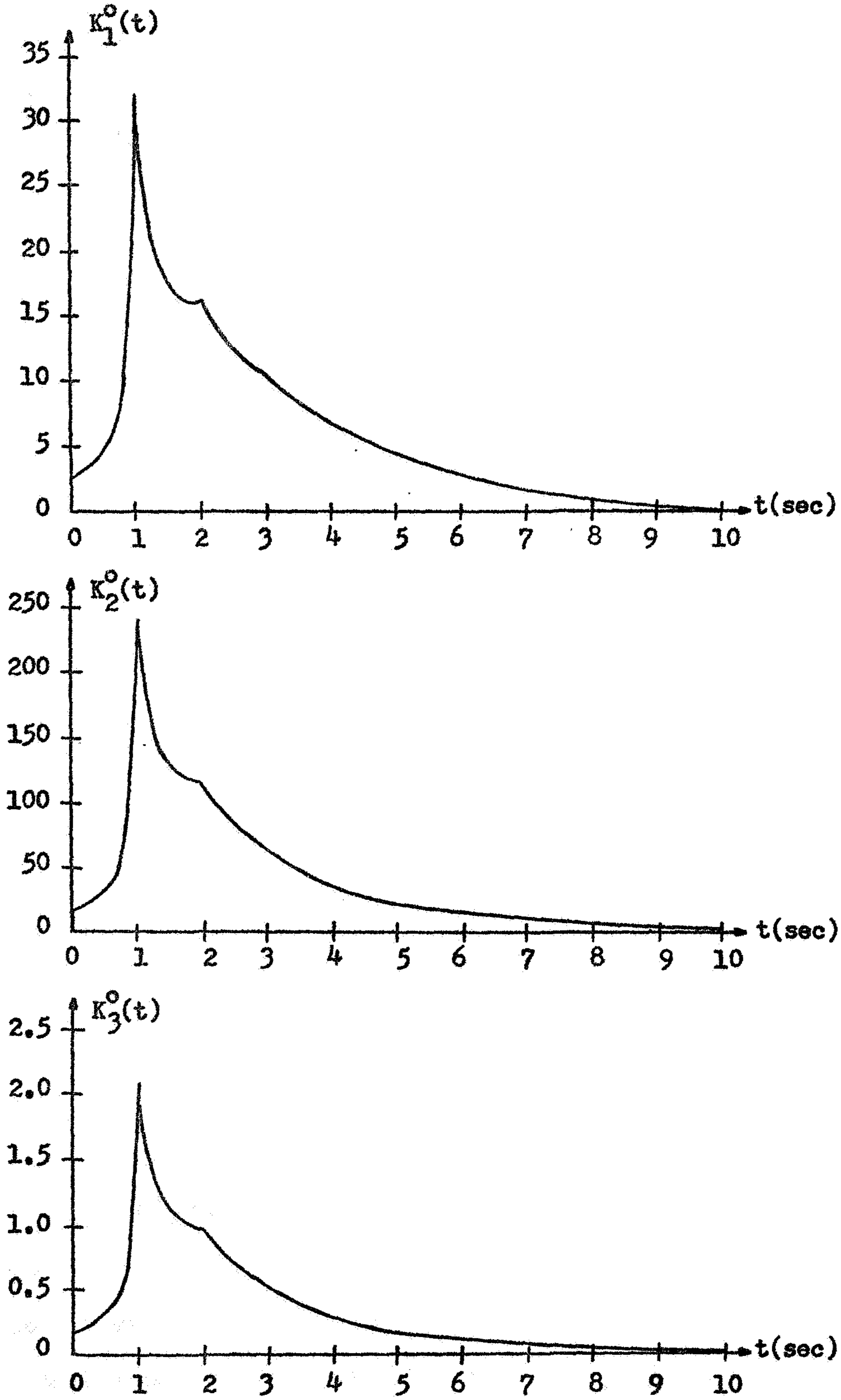


Figure 6.7 Optimal Feedback Coefficients-Minimum Norm Problem

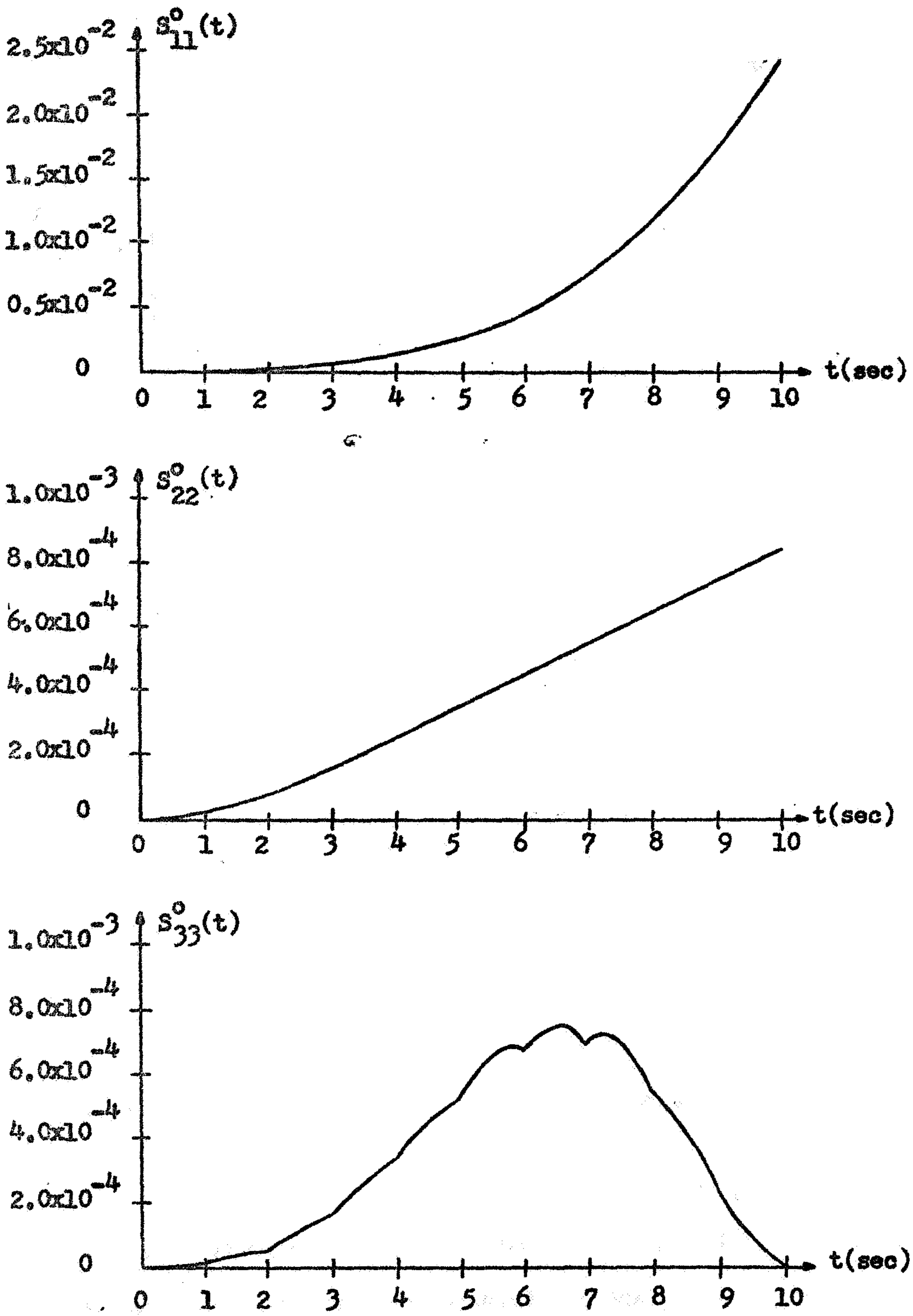


Figure 6.8 Optimal Response Covariances - Minimum Norm Problem

[6.1]. A brief outline of the derivation is given in Appendix E, along with numerical values of the coefficients in the equations. The booster equations have been linearized about some nominal (no-wind) trajectory. Other assumptions are that the vehicle is a rigid body, and that fuel-slosh and engine-inertia effects can be ignored.

The vehicle equations involving drift and pitch from the nominal trajectory are of the form (with the time dependence suppressed):

$$\ddot{y} = c_1\dot{y} + c_2\phi + c_3\dot{\phi} + c_4\beta + c_5\eta_1 + c_6\eta_2 \tag{6-15}$$

$$\ddot{\phi} = c_7\dot{y} + c_8\phi + c_9\dot{\phi} + c_{10}\beta + c_{11}\eta_1 + c_{12}\eta_2 \, , \tag{6-16}$$

where y is vehicle drift from the nominal trajectory, ϕ is pitch angle from nominal, β is the engine gimbal angle, the c_i's are given time-varying coefficients, and the dots indicate differentiation with respect to time. The time interval considered is from launch at t_o=0 seconds to booster burnout at T = 150 seconds. The initial conditions on the above equations are:

$$y(0) = \dot{y}(0) = \phi(0) = \dot{\phi}(0) = 0 \, , \tag{6-17}$$

since the initial perturbations from nominal are zero.

The variables η_1 and η_2 in (6-15) and (6-16) represent wind loadings on the vehicle, and are found by solving the following "wind-loading" equations:

$$\dot{\eta}_1 = c_{13}\eta_1 + c_{14}\omega_1 \qquad (6\text{-}18)$$

$$\dot{\eta}_2 = c_{15}\eta_2 + c_{16}\eta_3 + c_{17}\omega_1 \qquad (6\text{-}19)$$

$$\dot{\eta}_3 = c_{18}\eta_2 + c_{19}\omega_1 \ . \qquad (6\text{-}20)$$

where the c_i's are again time-varying coefficients. The ω_1 term is the output of a filter which models the incident winds, and which is described by:

$$\dot{\omega}_1 = c_{20}\omega_2 + c_{21}n \qquad (6\text{-}21)$$

$$\dot{\omega}_2 = c_{22}\omega_1 + c_{23}\omega_2 + c_{24}n \ . \qquad (6\text{-}22)$$

The filter equations above are driven by $n(t)$, a white Gaussian noise input which has zero mean and variance given by:

$$E[n(t)n(\tau)] = \delta(t-\tau) \ , \qquad (6\text{-}23)$$

where $\delta(t-\tau)$ denotes the Dirac delta function at $t = \tau$.

In this booster model, the control u is a scalar which drives the gimballed engines. The equation describing the gimbal actuator dynamics is assumed to be:

$$\dot{\beta} = -c_{25}\beta + c_{25}u \ . \qquad (6\text{-}24)$$

The initial values of η_1, η_2, η_3, ω_1, ω_2, and β in the above equations are all assumed to be zero.

Now define a 10-dimensional state vector $x=[y \ \dot{y} \ \phi \ \dot{\phi} \ \beta \ \omega_1 \ \omega_2 \eta_1 \eta_2 \eta_3]'$, and split the second-order equations (6-15) and (6-16) into two first-order ones. Then it can be seen that equations (6-15), (6-16), (6-18) to (6-20), (6-21), (6-22), and (6-24) can be put into the form of equation (2-1), with the dimensions $n = 10$ and $m = 1$, and with $x(0) = 0$.

The responses to be controlled are chosen to be:

$$r_1 = y \tag{6-25}$$

$$r_2 = \dot{y} \tag{6-26}$$

$$r_3 = \xi = c_{26}\dot{y} + \phi + c_{27}\omega_1 , \tag{6-27}$$

$$r_4 = \beta \tag{6-28}$$

$$\begin{aligned} r_5 = I_b = c_{28}\dot{y} + c_{29}\phi + c_{30}\dot{\phi} + c_{31}\beta \\ + c_{32}\eta_1 + c_{33}\eta_2 . \end{aligned} \tag{6-29}$$

The drift y, the drift rate $\dot{y}$, and the angle-of-attack ξ are of interest because they are measures of the error in the booster trajectory at burnout. The gimbal angle β is constrained by physical limitations to be less than five degrees during the flight. The response I_b is the bending-moment on the vehicle at a chosen point along the booster. This bending moment must be constrained so that vehicle strength limits will not be exceeded during the flight. The first three responses are actually of interest only at the end of the flight, while $r_4 = \beta$ and $r_5 = I_b$ must be controlled throughout the flight.

It will be shown in the discussion on the performance index that the derivatives of the latter two responses are also of interest. So define two more responses:

$$r_6 = \dot{\beta} = - c_{25}\beta + c_{25}u \tag{6-30}$$

$$r_7 = \dot{I}_b = c_{34}\dot{y} + c_{35}\phi + c_{36}\dot{\phi} + c_{37}\beta + c_{38}\omega_1 + c_{39}\eta_1 + c_{40}\eta_2 + c_{41}\eta_3 + c_{42}u \; . \tag{6-31}$$

More detailed expressions for all the above responses are given in Appendix E. In light of the definition of the state vector x and the control u, it can be seen that equations (6-25) to (6-31) can be written in the form of (2-3), with the dimension $\ell = 7$.

For this problem, it is assumed that perfect measurements of the state vector x are available. So the measurement vector z is

$$z(t) = x(t), \tag{6-32}$$

and there is no estimation problem. The control u will then be of the form

$$u = - K(t)x(t), \tag{6-33}$$

where K satisfies the properties in (2-9).

The performance index to be minimized is Skelton's "upper bound on the probability of mission failure" mentioned in Chapter 1 and derived in [6.1]. An outline of the derivation is given in Appendix F. The index is formed by first assigning an "error bound" γ_i to each response, r_i. Then the event of "mission failure" is defined to occur when any one of the responses exceeds its bound. An upper bound to the probability of "mission failure" is derived in terms of the response covariances, and becomes the performance index J_s:

$$J_s = g_1[S(T)] + \int_{t_o}^{T} g_2[S(t)]dt, \tag{6-34}$$

where S is defined in (2-17). For this example

$$g_1[S(T)] = \sum_{i=1}^{3} 2\,\Phi_N\left(-\frac{\gamma_i}{\sqrt{S_{ii}(T)}}\right), \tag{6-35}$$

$$g_2[S(t)] = \sum_{i=4}^{5} 2\,P_i(t), \tag{6-36}$$

where $$\Phi_N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2}\,dy, \tag{6-37}$$

$$P_i(t) = \frac{\exp\left[-\gamma_i^{\,2}/2S_{ii}\right]}{\sqrt{2\pi}\,S_{ii}} \left\{ \frac{\sigma_i \exp\left[-\rho_i^{\,2}/2\,\sigma_i^{\,2}\right]}{\sqrt{2\pi}} - \rho_i\left[1 - \Phi_N\left(\frac{\rho_i}{\sigma_i}\right)\right] \right\}, \tag{6-38}$$

and $$\rho_i = -\frac{\gamma_i S_{ij}}{S_{ii}}, \quad \sigma_i = \left[S_{jj} - \frac{S_{ij}^{\,2}}{S_{ii}}\right]^{1/2}, \tag{6-39}$$

for j = i+2. By referring to the derivation of J_s in Appendix F, it can be seen that the response vector r defined in (6-25) to (6-31) is in the proper form for use in J_s. That is, the terminal responses are formed first, with the "in-flight" responses following. As is also mentioned in Appendix F, the responses $r_6=\dot{\beta}$ and $r_7=\dot{I}_b$ are not bounded, but are used in the evaluation of how often $r_4=\beta$ and $r_5=I_b$ exceed their bounds.

It can be seen that the J_s performance index is a special case of the general performance index J in (2-16). The problem to be solved is then the same as the general problem discussed in Section 2.2, using the above system, response, and performance index equations. It is to find the $u \in U$ (defined in (2-9)) such that J_s in (6-34) is minimized, subject to the system equations defined above.

6.3.2 A Suboptimal Problem

The original intention in this example was to apply both the PGM and DGIM algorithms directly to the problem of minimizing J_s. When the PGM algorithm was applied, however, the following difficulty arose. As shown in Figure 5.3, the second step in PGM (after choosing an initial $\hat{s}_o \in \alpha$) is to compute the gradient vector $DJ_s(\hat{s}_o)$ and use it to specify the first J_Q-problem. The gradient vector $DJ_s(\hat{s}_o)$ is found (see (3-20)) in this example by first computing the partial derivative matrices $\frac{\partial g_1}{\partial S}$ and $\frac{\partial g_2}{\partial S}(t)$. For the several initial points $\hat{s}_o$ tried, it was found that the elements of the matrix $\frac{\partial g_2}{\partial S}(t)$ were smaller than 10^{-100} for $t \in [145,150]$, and for most of the $\hat{s}_o$ tried were less than 10^{-50} for $t \in [130,150]$. Now, the next step in PGM is that of setting $Q_F(T) = \frac{\partial g_1}{\partial S}$ and $Q(t) = \frac{\partial g_2}{\partial S}(t)$, and using the Q's in the backward Riccati equation (2-23) to get $K_o{}^*(t)$ by (2-22). To solve the Riccati equation, the inverse of the matrix $D'(t)Q(t)D(t)$ must be computed for values of t in the entire time interval $[0,150]$. But this computation could not be performed, due to the extremely small size of the elements of $Q(t) = \frac{\partial g_2}{\partial S}(t)$, as described above. An attempt was made to approximate $\frac{\partial g_2}{\partial S}(t)$ by a matrix function $\frac{\widetilde{\partial g_2}}{\partial S}(t)$ whose corresponding elements were somewhat larger (greater than 10^{-20}). Several approximations were tried, but when $Q(t) = \frac{\widetilde{\partial g_2}}{\partial S}(t)$ was used in the Riccati equation, the numerical integration went unstable for every approximation.

In light of these difficulties, the attempt to use PGM to minimize J_s directly was abandoned. Instead, another performance index, J_N, was formed, and the PGM algorithm was used to minimize it. The J_N index was chosen heuristically such that it "matched" the properties of J_s in some sense, and such that the above difficulties with its gradient $DJ_N(\hat{s})$ would not be encountered. It was noticed that the dominant terms in the J_s index varied as $\frac{S_{ii}}{\gamma_i^2}$. Therefore, J_N was chosen to be:

$$J_N = h_1[S(T)] + \int_{t_o}^{T} h_2[S(t)]dt \; , \tag{6-40}$$

where
$$h_1[S(T)] = \tfrac{1}{2} \sum_{i=1}^{3} \left[\frac{S_{ii}(T)}{\delta_i^{\,2}}\right]^2 \tag{6-41}$$

$$h_2[S(t)] = \tfrac{1}{2} \sum_{i=4}^{7} \left[\frac{S_{ii}(t)}{\delta_i^{\,2}}\right]^2 . \tag{6-42}$$

The δ_i's are given real positive numbers which weight the various covariances, and tend to equalize the discrepancy in magnitude between, say, the gimbal angle and bending moment covariances.

The J_N-problem is then to find the $u \in U$ that minimizes J_N, subject to the system equations defined in section 6.3.1. Since the PGM algorithm will be applied to this problem, it is useful to check whether J_N satisfies the hypotheses in the theorems derived in Chapters 3, 4, and 5. To do this, note that J_N is a norm-type of performance index, and is very similar in form to the performance index used in the first example (see section 6.2, equation (6-6)). Thus the comments made in that section concerning the hypotheses of the theorems are also applicable to J_N. In particular, J_N is well-defined and convex, and its partial derivatives satisfy the hypotheses of Theorems 3.1 and 4.1. The question of existence of a solution to the J_N-problem is still unresolved, however.

It was found that the computational difficulties which were encountered when PGM was applied to J_s did not occur when PGM was applied to J_N. Using the δ_i's given in section 6.3.3, it was found that the elements of $\frac{\partial h_2}{\partial S}(t)$ were large enough so that the problems mentioned above were avoided.

There are two objectives in using the PGM algorithm to minimize J_N. The first is to see if PGM can be applied to a problem with a high-order, time-varying set of system equations. These equations, together with the admissible control set U, define a "set of attainability" α that is considerably more complex than the one in Example 1. Thus a successful application of PGM would indicate that it can be used to solve practical problems, which usually have complex dynamical models. The second objective is to use PGM on J_N to obtain "good" controls for Skelton's gust-alleviation problem. The quality of the controls will be measured by J_s, Skelton's "upper-bound" performance index. If the controls are of good quality, the example would demonstrate the usefulness of PGM (and the supporting function-space theory) in a specific practical application.

6.3.3 Results and Discussion

The PGM and DGIM algorithms described in Chapter 5 were applied to the load-relief problem in the following way. PGM was applied to the problem of minimizing J_N; the resulting sequence of points $\{\hat{s}_i\}$ in the "set of attainability" α were stored, and later were evaluated using the J_s performance index. The DGIM algorithm was applied directly to the problem of J_s, and the results obtained using this iterative method were compared to those obtained by using PGM. Some of the computer techniques and descriptions of the programs which implement the algorithms are given in Appendix G.

The specific PGM algorithm applied in minimizing J_N is shown in Figure 6.9, and the DGIM algorithm used to minimize J_s is shown in Figure 6.10. These algorithms are similar to those used in the first example in Section 6.2, with the following exceptions:

1) The initial feedback coefficient $K_o(t)$ was found by choosing an initial quadratic problem specified by $\hat{q}_{IC}$, and then solving the Riccati equation in (2-23) for $K_o(t)$. This initial feedback coefficient, when used in the system equations, then defined the starting point, $\hat{s}_o \in \alpha$, for the iterations.

2) The weighting factor λ in the DGIM algorithm was selected beforehand, and kept constant throughout the iteration sequence.

3) No stopping condition was invoked, as was done in the first example, due to the high cost in computational time of each iteration. Instead, the algorithms were allowed to run until "good" controls resulted, or until a clear pattern of the sequence of iterations emerged.

4) As mentioned above, the PGM algorithm was applied to the J_N performance index, and the DGIM algorithm to J_s. It was only afterward that the two sequences of points $\{\hat{s}_i\}$ (in PGM) and $\{\hat{s}_i{}^*\}$ (in DGIM) were compared on the common basis of Skelton's J_s performance index. This contrasts with the first example in this Chapter, in which both algorithms were used to minimize the same performance index.

Two separate iteration sequences were run for the load-relief

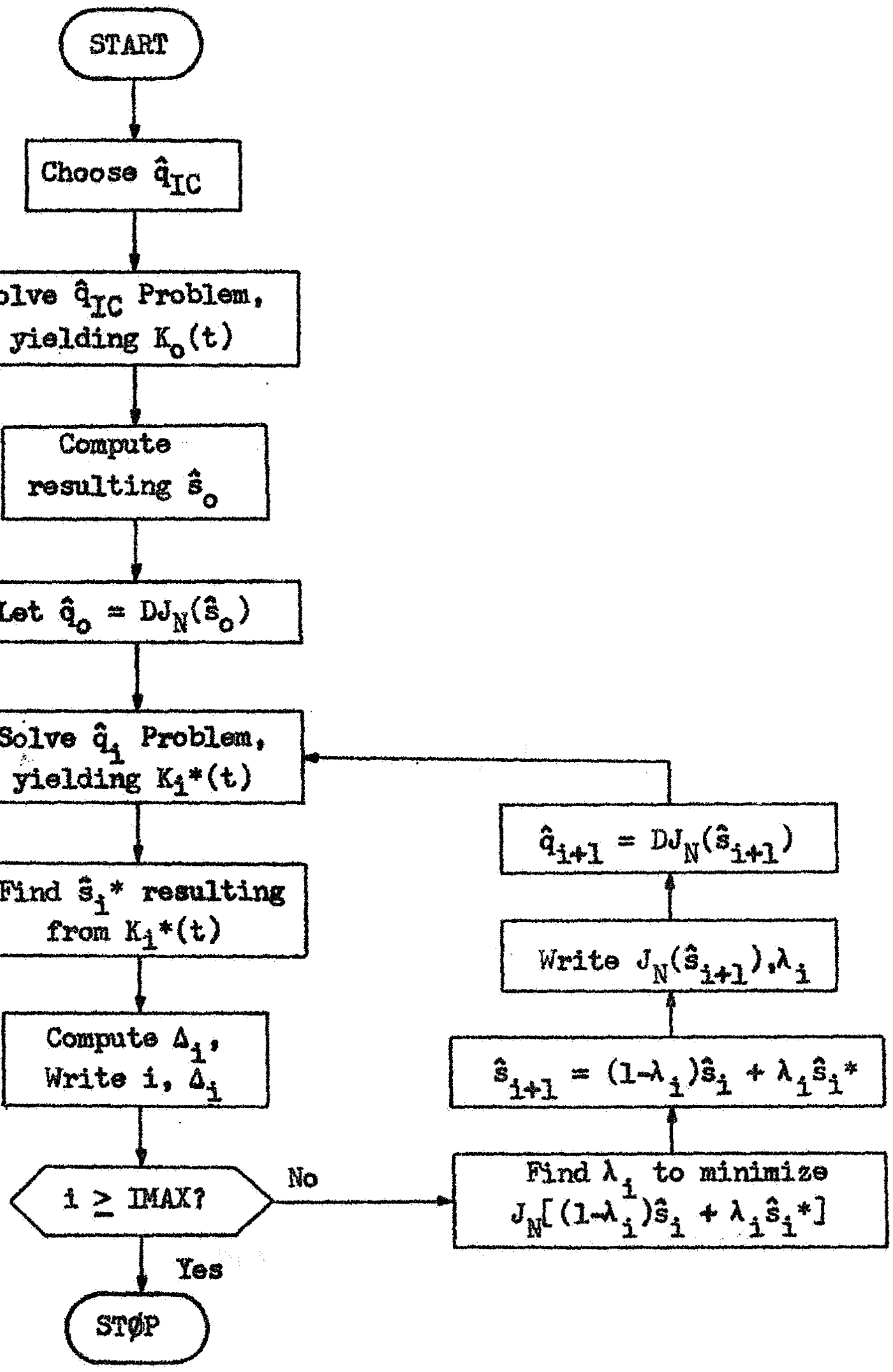


Figure 6.9 PGM Algorithm, Load - Relief Problem

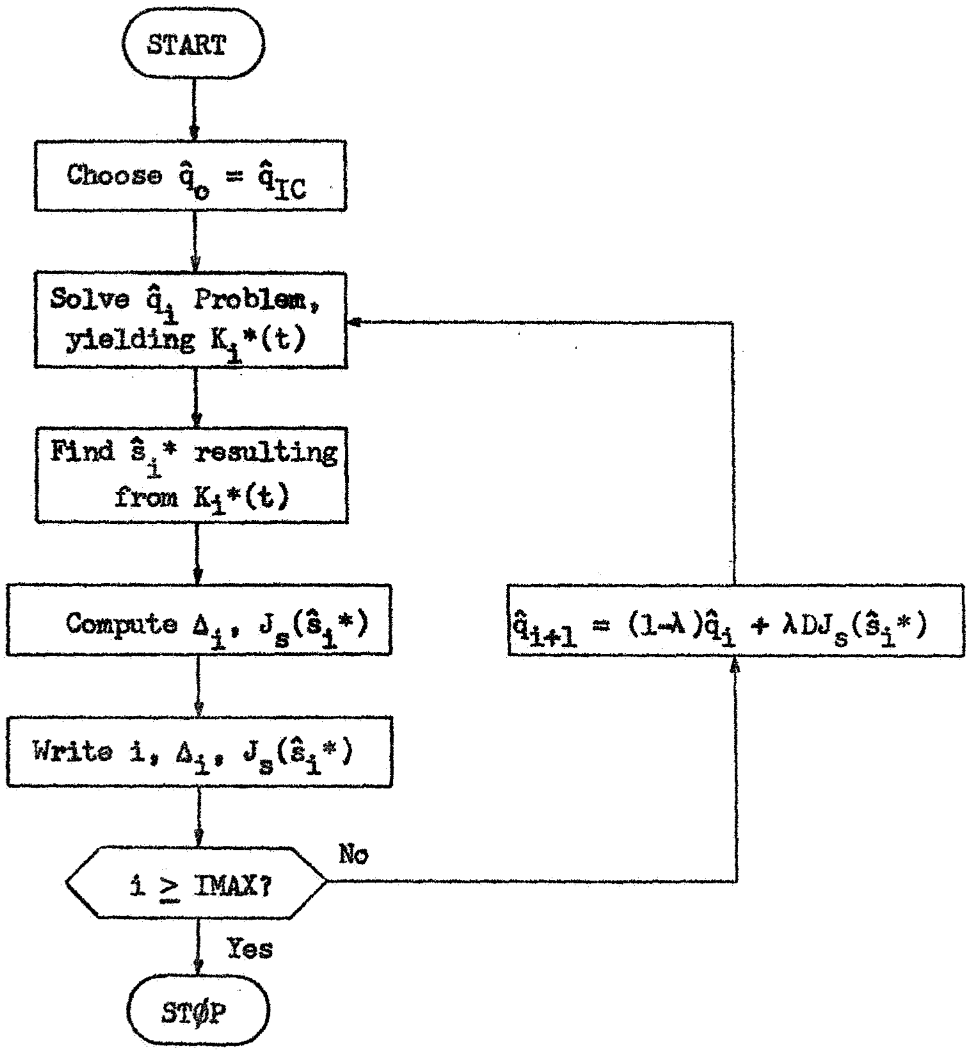


Figure 6.10 DGIM Algorithm, Load - Relief Problem

example. The differences between the two sequences were in a) the initial feedback coefficient, $K_o(t)$; b) the "error bounds" γ_i in J_s; c) the values of δ_i in J_N; and d) the value of the weighting factor λ used in the DGIM algorithm. The results of the two iteration sequences are presented as follows:

Iteration Sequence 1

In this sequence, the values of γ_i used in J_s and of δ_i used in J_N are as follows:

$$\gamma_1 = 3000 \qquad \delta_1 = 3.0 \times 10^{3}$$

$$\gamma_2 = 40 \qquad \delta_2 = 0.4$$

$$\gamma_3 = 8.73 \times 10^{-2} \qquad \delta_3 = 1.0 \times 10^{-4}$$

$$\gamma_4 = 8.73 \times 10^{-2} \qquad \delta_4 = 1.0 \times 10^{-4}$$

$$\gamma_5 = 2.25 \times 10^{6} \qquad \delta_5 = 5.0 \times 10^{9}$$

$$\delta_6 = 2.0 \times 10^{-4}$$

$$\delta_7 = 1.0 \times 10^{10}$$

Remember that the responses r_6 and r_7 are not bounded, but are used in determining how often r_4 and r_5 exceed their bounds. The error bounds γ_i chosen are similar to those used by Skelton in [6.1], and are motivated by practical considerations. Several values of the δ_i's were tried; the ones listed above gave reasonable results. The value of λ used in DGIM was $\lambda = 0.01$. Larger values of λ were tried (0.9, 0.8, 0.1), but when used in DGIM they produced $\hat{q}_i$'s that caused the backward numerical integration of the Riccati equation to go unstable in the first iteration.

The initial condition on the iteration sequence was the $\hat{q}_{IC}$-problem, which when solved by means of the Riccati equation in (2-23) yielded the initial feedback coefficient, $K_o(t)$. This $\hat{q}_{IC}$-problem is specified by the (7 x 7) quadratic coefficient matrices $Q_F(T)$ and $Q(t)$, $t\in[t_o,T]$. The terminal time coefficient matrix used was:

$$Q_F(T) = \begin{bmatrix} 1.1481\text{x}10^{-7} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.8500\text{x}10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.1288\text{x}10^{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-43}$$

The matrix function of time $Q(t)$ which was used is specified by:

$$Q(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{44} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{77} \end{bmatrix}, \tag{6-44}$$

where the values of Q_{ii}, $i = 4,5,6,7$, at 5-second intervals of time are given in Table 6.1. The values of the Q_{ii} between the points given were found by linear interpolation. The above values of $Q_F(T)$ and $Q(t)$ were chosen rather arbitrarily; it was found that the resulting feedback coefficient $K_o(t)$ was not an especially "good" one as measured by the J_s-performance index, and thus it was felt that the $Q_F(T)$ and $Q(t)$

Table 6.1

Initial Values of Q for Iteration Sequence 1

t (sec)	Q_{44}	Q_{55}	Q_{66}	Q_{77}
0	2.5000E+01	7.3272E-15	4.3023E+04	1.9702E-13
5	1.9017E+01	5.4384E-15	2.4365E+02	1.1464E-15
10	3.1138E+00	7.8732E-16	1.8596E+02	1.0724E-15
15	7.4873E-01	1.0823E-15	1.1141E+02	3.9195E-16
20	2.4017E+00	9.6928E-15	3.1181E+01	2.0689E-15
25	5.5218E+00	2.2607E-14	1.0846E+02	7.7383E-15
30	1.1802E+01	1.6734E-14	2.9907E+02	1.2043E-13
35	1.4041E+01	5.9144E-14	5.6583E+02	1.8278E-13
40	2.7390E+01	9.4812E-14	6.4067E+02	6.0625E-13
45	9.2457E+01	1.1030E-13	2.2821E+02	2.6193E-12
50	1.1998E+02	2.0463E-13	5.3575E+02	3.3948E-12
55	1.1028E+02	3.2521E-13	2.8091E+03	4.1985E-12
60	6.9697E+01	4.2280E-13	2.9524E+03	4.3660E-12
65	2.3853E+01	4.8940E-13	2.6012E+03	4.3048E-12
70	3.4025E+01	4.5760E-13	1.6918E+03	2.8500E-12
75	9.5223E+01	4.4720E-13	1.7380E+03	1.9522E-12
80	7.0470E+01	1.7850E-13	1.2672E+03	2.5013E-12
85	4.2655E+01	6.5432E-14	1.0198E+03	3.4095E-12
90	2.2162E+01	2.6492E-14	4.8433E+02	1.6904E-13
95	2.0913E+01	3.9351E-14	3.2536E+02	1.8183E-13
100	3.8330E+01	2.0048E-14	2.1248E+02	1.2071E-13
105	7.1838E+01	5.0808E-15	1.0425E+02	3.1497E-14
110	8.8582E+01	6.7960E-15	1.4293E+02	3.2265E-14
115	4.8690E+01	1.8488E-14	7.2774E+02	4.2230E-14
120	7.9978E+00	2.5748E-14	1.2272E+03	6.5868E-14
125	1.5176E+01	3.0730E-14	7.3791E+02	2.9688E-14
130	1.7215E+01	8.4028E-15	1.5854E+02	2.7523E-14
135	8.7663E+01	9.9508E-15	4.4584E+03	8.8748E-14
140	1.4634E+02	1.7084E-13	1.7120E+04	1.7169E-13
145	2.7420E+03	2.1416E-12	1.5745E+05	1.5055E-12
150	8.7012E+02	7.3268E-13	1.2861E+00	6.3228E-15

matrices chosen were realistic initial guesses.

The results of applying PGM to the problem of minimizing J_N are shown in Figure 6.11. The values of J_N are plotted with respect to computer time on a CDC 6500 computing system, using Fortran IV. The computing time for each iteration of PGM was about 13.2 minutes. This time included the numerical integration of the Riccati equation in (2-23) and the response covariance equations in (3-1) and (3-3), as well as the process of finding λ_i to minimize J_N shown in Figure 6.9. Each point in Figure 6.11 represents an iteration. It can be seen that the successive values of J_N decrease monotonically, which is to be expected from the nature of the PGM algorithm.

The sequence of points $\{\hat{s}_i\}$, $i = 0,1,\ldots,5$, which resulted from the above application of PGM, were then evaluated in Skelton's J_s performance index, as shown in Figure 6.12. (See Appendix G for details on how this evaluation was carried out.) In addition, the results of applying the DGIM algorithm to minimizing J_s are also shown in Figure 6.12. In this figure, the values of J_s produced by PGM do not decrease monotonically for the last two iterations. This is due, in part, to the "mismatch" between the performance indices J_N and J_s. Apparently, however, there is significant correspondence between J_N and J_s, because the last three iteration points produce values of J_s on the order of 5×10^{-6}. As an upper bound to the probability of mission failure, this figure shows that the system performance is quite good for these points. This is especially true when the J_s values are compared to the initial J_s value of 0.0302. The DGIM algorithm also shows a decrease in J_s, but this decrease is not as substantial. It should be noted that one iteration using the DGIM algorithm on J_s took about 14.7 minutes of

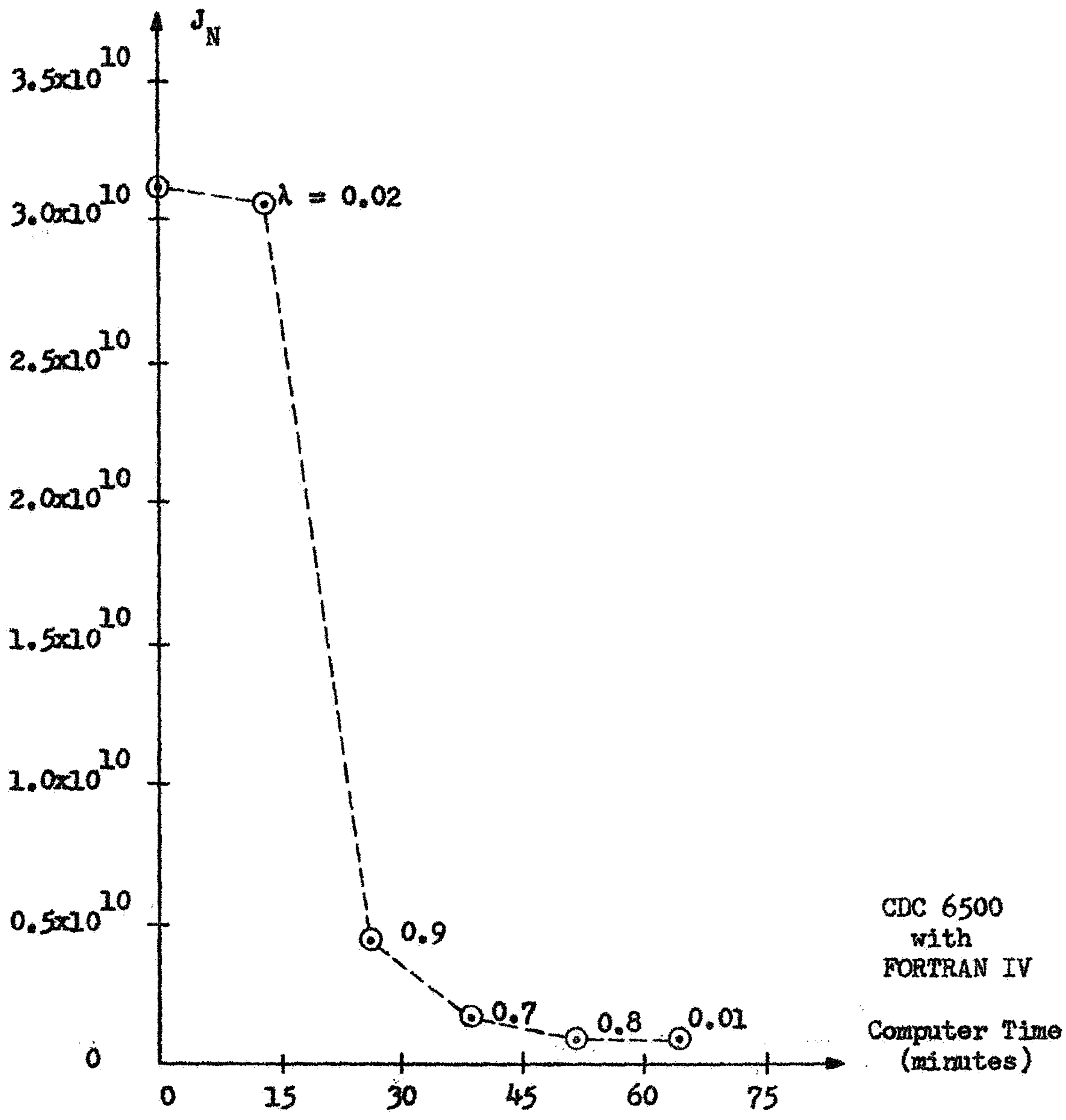


Figure 6.11 J_N in Load - Relief Problem, Sequence 1

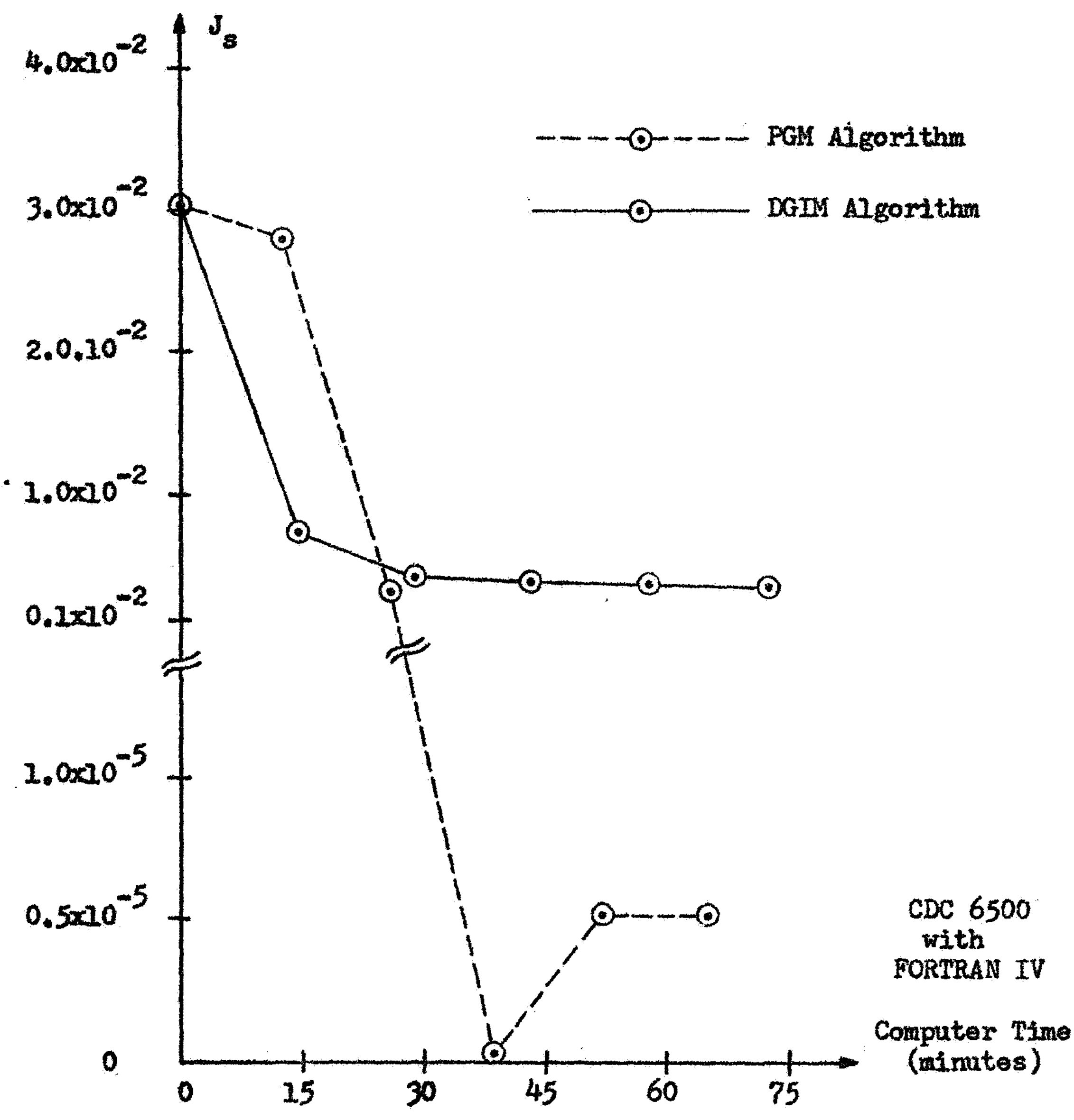


Figure 6.12 J_s in Load - Relief Problem, Sequence 1

computer time. This is longer than the PGM iteration time mentioned above, but this is due only to the fact that it took more time to evaluate the J_s performance index than it did to evaluate J_N. The process of choosing $\hat{q}_{i+1}$ in DGIM actually took less time (~ 10 sec.) than did the process of choosing $\hat{s}_{i+1}$ in PGM (~ 30 sec.).

By referring to the definition of PGM in Figure 6.9, it can be seen that there is a practical difficulty in making use of the results plotted in Figure 6.11. This difficulty is that the controls which produce the sequence $\{\hat{s}_i\}$, $i = 1,2,\ldots 5$ are not known. However, there <u>is</u> a known sequence of feedback coefficients produced by PGM; namely, the sequence $\{K_i{}^*(t)\}$, $i = 1,2,\ldots 5$. This sequence results from solving the associated $\hat{q}_i$-problems in the algorithm. Since these coefficients define the practical controls of interest, it is useful to evaluate the sequence of points $\{\hat{s}_i{}^*\}$, $i = 1,2,\ldots 5$ (produced by the coefficients $\{K_i{}^*(t)\}$) using the J_s performance index. Note that the $\hat{s}_i{}^*$ are points on the boundary of α. The results of this evaluation are shown in Figure 6.13. This figure shows that the last three feedback coefficients in the sequence define very good controllers, because they produce a probability of mission failure that is less than 10^{-5}. In fact, using $K_3{}^*(t)$ in the load-relief controller produces a probability of mission failure less than 10^{-8}. So Figure 6.13 shows that using the PGM algorithm on J_N to produce load-relief controllers is very useful from a practical point of view.

The numerical values of the last computed feedback coefficient, $K_5{}^*(t)$, are given at 5-second intervals in Table 6.2. The superscripts for the K's denote vector components, and the intermediate values not in

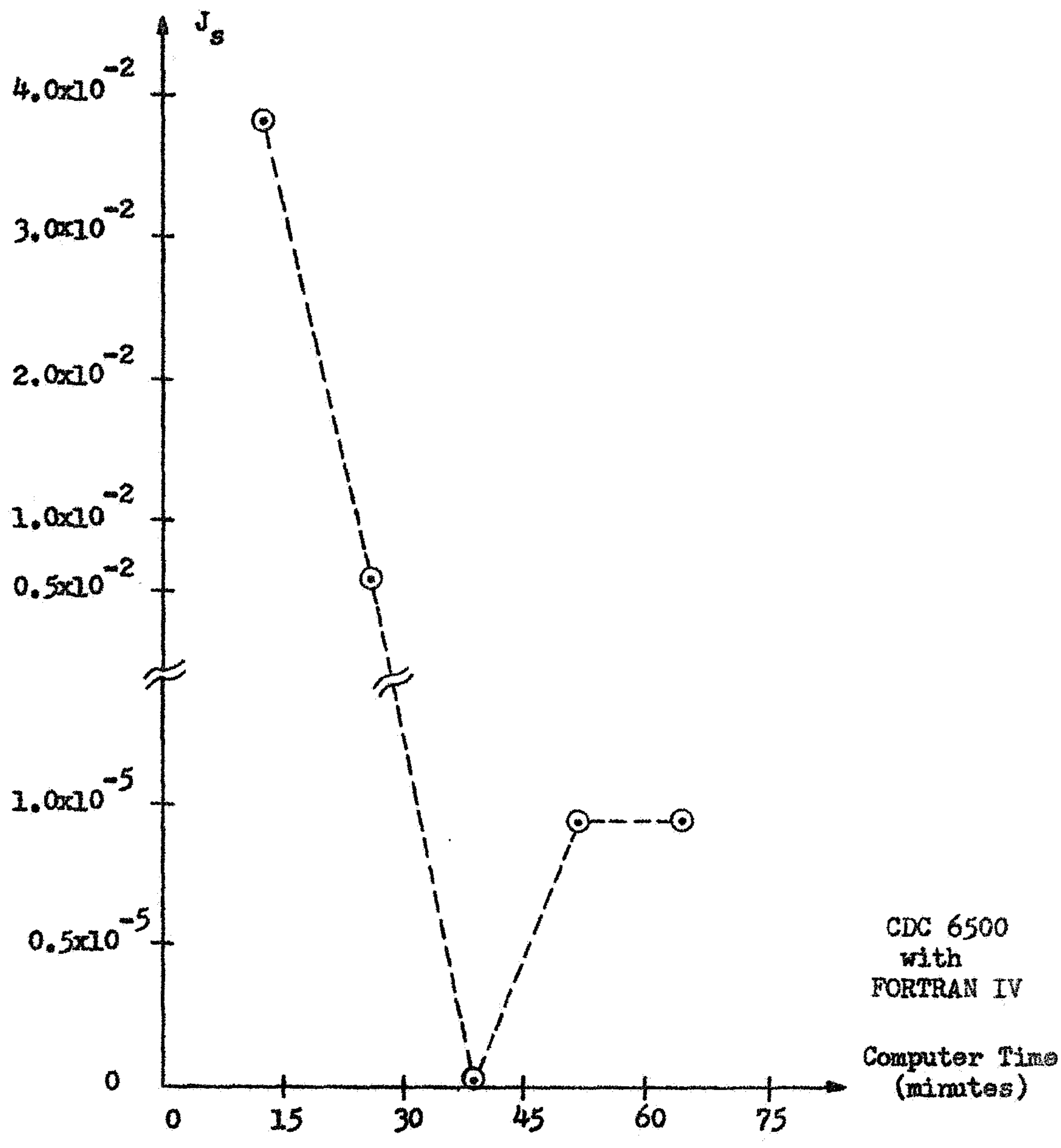


Figure 6.13 $J_s(\hat{s}_i^*)$ found by PGM, Sequence 1

Table 6.2

Values of $K_5^*(t)$ in Iteration Sequence 1

t (sec)	K^1	K^2	K^3	K^4	K^5
0	-1.967161E-09	-1.577056E-05	-4.775135E-03	-4.365209E-02	-9.375968E-01
5	-6.828551E-09	-1.059880E-04	-2.775308E-02	-2.208662E-01	-7.784554E-01
10	-9.633210E-09	-2.298986E-04	-3.878352E-02	-2.557005E-01	-5.765095E-01
15	-4.243314E-08	-5.161796E-04	-9.644079E-02	-5.108776E-01	-6.177454E-01
20	-8.329531E-08	-5.209792E-04	-8.464142E-02	-4.054240E-01	-6.922881E-01
25	-1.151047E-07	-3.662190E-04	-5.867649E-02	-2.675463E-01	-8.201857E-01
30	-1.875442E-07	-3.342301E-04	-5.569143E-02	-2.595005E-01	-8.191881E-01
35	-2.501043E-07	-2.475459E-04	-4.700241E-02	-2.595724E-01	-7.964882E-01
40	-1.370570E-07	-4.926201E-05	-1.806784E-02	-1.657271E-01	-8.356232E-01
45	-4.420061E-08	2.107988E-05	-6.814281E-03	-1.000516E-01	-8.866550E-01
50	-2.267410E-08	5.726257E-05	-9.808739E-03	-1.115150E-01	-8.835587E-01
55	-1.328083E-08	7.888790E-05	-1.447105E-02	-1.270963E-01	-8.770844E-01
60	-1.036223E-08	6.873967E-05	-1.501017E-02	-1.343835E-01	-8.688651E-01
65	-1.171283E-08	4.372994E-05	-1.106689E-02	-1.407995E-01	-8.597094E-01
70	-2.777984E-08	1.632863E-05	-6.276819E-03	-1.702765E-01	-8.270475E-01
75	-9.133763E-08	-5.698385E-05	2.690068E-02	-1.731755E-01	-7.565647E-01
80	-1.342217E-07	-5.511327E-05	2.398689E-02	-8.024009E-02	-8.321232E-01
85	-9.738237E-08	-2.731841E-05	1.163417E-02	-2.187948E-02	-9.162273E-01
90	-7.503625E-07	-9.730465E-05	3.706178E-02	-2.183641E-01	-6.061720E-01
95	-6.013327E-07	-6.067767E-05	2.251845E-02	-1.486264E-01	-6.952174E-01
100	-6.819860E-07	-5.763611E-05	2.090442E-02	-1.260969E-01	-6.686777E-01
105	-2.348645E-06	-1.099180E-04	1.087570E-02	-3.331676E-01	-3.448450E-01
110	-3.210469E-06	-1.231351E-04	-2.126152E-03	-3.352417E-01	-3.194828E-01
115	-2.386843E-06	-8.362025E-05	-1.369859E-03	-1.831933E-01	-4.861909E-01
120	-1.159404E-06	-3.249426E-05	-6.800029E-03	-8.269107E-02	-7.868615E-01
125	-1.238613E-06	-3.981824E-05	-1.214634E-02	-1.053441E-01	-7.419552E-01
130	-1.352620E-06	-7.624554E-05	-3.057597E-02	-1.783781E-01	-6.533101E-01
135	-6.813620E-07	-9.366808E-05	-3.429415E-02	-1.382658E-01	-7.367429E-01
140	-1.817919E-07	-4.112413E-04	-1.907707E-01	-8.993201E-01	-2.397103E-01
145	-2.821534E-09	-1.049919E-04	-2.335287E-02	-6.238737E-02	-7.893030E-01
150	0.	2.671876E-23	0.	3.1[illegible]0476E-04	-9.975746E-01

Table 6.2 (cont'd.)

t (sec)	K^6	K^7	K^8	K^9	K^{10}
0	-5.411534E-05	1.491794E-01	-3.224725E-04	1.823202E-04	-7.258457E-04
5	-4.021550E-04	1.128458E+00	-2.015605E-03	1.156453E-03	-5.984607E-03
10	2.614783E-05	2.181520E+00	-1.780627E-03	2.418299E-03	-8.240015E-03
15	1.176048E-03	6.839948E+00	-4.569403E-03	3.226925E-03	-1.760300E-02
20	1.662556E-03	8.016831E+00	-3.439169E-03	3.168694E-03	-1.834149E-02
25	1.384723E-03	6.905946E+00	-4.443621E-03	-1.015586E-03	-1.918594E-02
30	1.193956E-03	7.472749E+00	-6.295983E-03	-3.567425E-03	-2.582695E-02
35	9.738169E-05	6.609259E+00	-1.108961E-02	-9.310522E-03	-4.277537E-02
40	-1.409071E-03	1.810289E+00	-2.088991E-02	-2.228024E-02	-6.066281E-02
45	-1.903702E-03	-3.590090E-01	-3.750610E-02	-4.085537E-02	-8.344293E-02
50	-2.458313E-03	-1.118772E+00	-5.750837E-02	-5.963600E-02	-1.157881E-01
55	-3.003534E-03	-1.603106E+00	-8.233633E-02	-8.096545E-02	-1.529093E-01
60	-3.264426E-03	-1.919681E+00	-1.122072E-01	-1.047463E-01	-1.942010E-01
65	-3.332959E-03	-2.191683E+00	-1.560166E-01	-1.380571E-01	-2.510521E-01
70	-3.572511E-03	-2.650484E+00	-2.029388E-01	-1.699649E-01	-3.134977E-01
75	-2.846339E-03	-2.328257E+00	-2.466055E-01	-1.965240E-01	-3.784778E-01
80	-5.700135E-04	-8.523870E-01	-2.616047E-01	-1.982130E-01	-3.423705E-01
85	3.653568E-04	-2.387017E-01	-2.573681E-01	-1.846398E-01	-2.938858E-01
90	-1.963663E-04	-8.714215E-01	-1.785398E-01	-1.212809E-01	-2.442611E-01
95	1.533592E-04	-4.747308E-01	-1.959048E-01	-1.263109E-01	-2.284952E-01
100	2.609344E-04	-2.958598E-01	-1.897462E-01	-1.162137E-01	-2.074761E-01
105	-2.456260E-05	-5.885368E-01	-1.511500E-01	-8.792257E-02	-1.851415E-01
110	6.910143E-05	-4.623230E-01	-1.412471E-01	-7.784697E-02	-1.594211E-01
115	2.377986E-04	-1.690771E-01	-1.276884E-01	-6.667158E-02	-1.220637E-01
120	2.286267E-04	-3.938015E-02	-9.694866E-02	-4.776391E-02	-7.736455E-02
125	1.819906E-04	-2.897360E-02	-9.109777E-02	-4.253463E-02	-6.950297E-02
130	1.219577E-04	-1.755641E-02	-8.054399E-02	-3.558458E-02	-5.981488E-02
135	7.483371E-05	-1.361762E-03	-6.887186E-02	-2.881630E-02	-4.696574E-02
140	7.650577E-06	-1.412853E-01	-5.002474E-02	-1.921045E-02	-4.090615E-02
145	1.635726E-05	1.293112E-03	-4.021600E-02	-1.509546E-02	-2.394303E-02
150	2.736001E-20	0.	-2.747016E-02	-9.767462E-03	-1.465025E-02

the table were found by linear interpolation. The feedback coefficient $K_5^*(t)$ was found by solving the J_Q-problem defined by the quadratic coefficient matrices $Q_F^5(T)$ and $Q^5(t)$, $t\in[0,150]$. The terminal time coefficient matrix was:

$$Q_F^5(T) = \begin{bmatrix} 8.5417\times10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.2420\times10^{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.6642\times10^{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

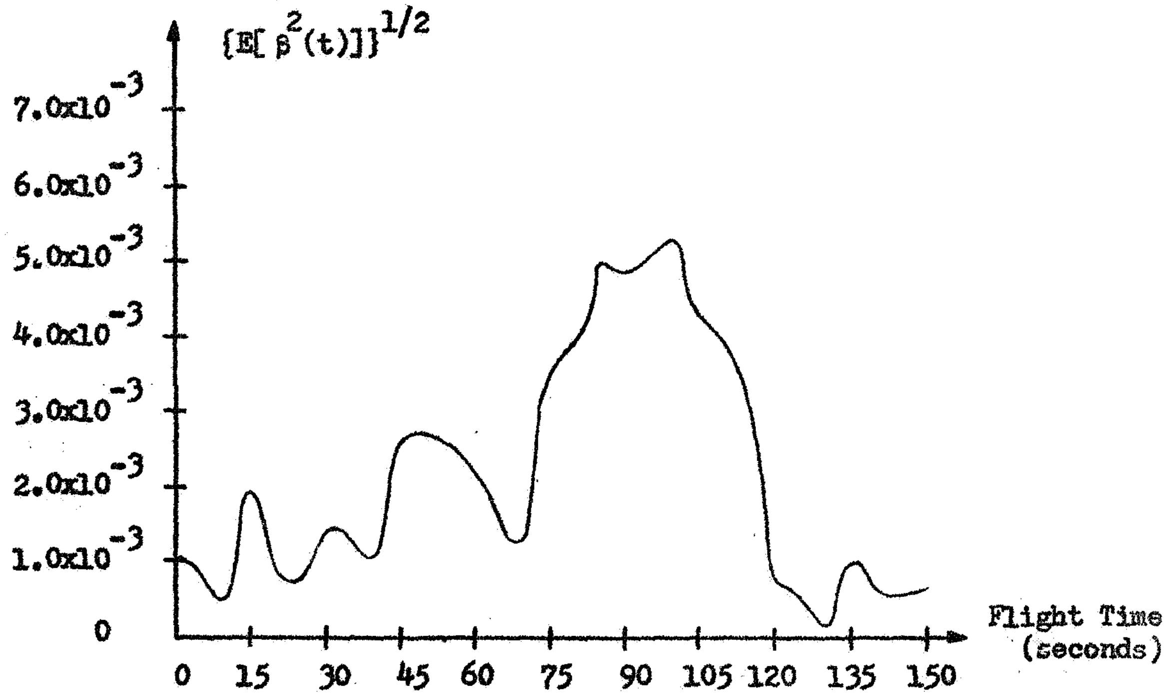
The time-varying coefficient matrix $Q^5(t)$ was of the same form as the Q-matrix in (6-44). The values of the diagonal elements of interest are given in Table 6.3 at 5-second intervals of time. Again, the intermediate values of the elements were found by linear interpolation.

The standard deviations of the "in-flight" responses, $r_4 = \beta$ and $r_5 = I_b$, which resulted when $K_5^*(t)$ was used in the covariance equations are plotted in Figure 6.14 as a function of time. (Remember that the responses are zero-mean Gaussian random variables; thus the response statistics are completely specified by the standard deviations.) From the figure, the peak standard deviation of β is about 5.0×10^{-3}, and that of I_b is about 3.1×10^5. Since the corresponding "error bounds" on β and I_b are $\gamma_4 = 8.73 \times 10^{-2}$ and $\gamma_5 = 2.25 \times 10^6$, it can be seen that the probability that the responses exceed their error bounds at any given time is very small (certainly less than 10^{-6}). So $K_5^*(t)$ produces "good" in-flight responses. The standard deviations of the responses of interest at the terminal time were:

Table 6.3

Values of $Q^5(t)$ in Iteration Sequence 1

t (sec)	Q^5_{44}	Q^5_{55}	Q^5_{66}	Q^5_{77}
0	1.0000E+02	7.3272E-11	2.0455E+02	1.4989E-10
5	1.1379E+02	7.9686E-11	1.3343E+01	9.8453E-12
10	2.2121E+02	1.7002E-10	7.0264E+00	4.6958E-12
15	1.0043E+02	5.3046E-11	1.8865E+00	2.1891E-12
20	7.6030E+01	8.9346E-11	3.0507E+00	6.4388E-12
25	7.6062E+01	1.0874E-10	2.1771E+01	3.1328E-11
30	2.6233E+02	2.3367E-10	2.1762E+01	6.2435E-11
35	1.6051E+02	4.7609E-10	1.7141E+01	9.5066E-11
40	1.0058E+02	8.5148E-10	3.4799E+01	2.8659E-10
45	6.5517E+02	1.3081E-09	4.0404E+01	1.0095E-09
50	7.4129E+02	1.7570E-09	7.1948E+01	1.3809E-09
55	6.5610E+02	2.4032E-09	1.1589E+02	1.7108E-09
60	5.4190E+02	3.1689E-09	1.4612E+02	1.7897E-09
65	2.0509E+02	3.5429E-09	1.7334E+02	1.6003E-09
70	1.8970E+02	3.7522E-09	2.1980E+02	9.3317E-10
75	1.1401E+03	3.5391E-09	3.2926E+02	3.5197E-10
80	1.5850E+03	1.6680E-09	3.2659E+02	3.8297E-10
85	2.0374E+03	9.3219E-10	2.4220E+02	8.3950E-10
90	2.9384E+03	5.9877E-10	4.4397E+01	2.0864E-11
95	2.3959E+03	7.6845E-10	2.7765E+01	4.4546E-11
100	1.9223E+03	8.7642E-10	1.1463E+01	4.0210E-11
105	1.5927E+03	4.9296E-10	6.5179E+00	5.7485E-12
110	1.0346E+03	4.9865E-10	5.8768E+00	5.4617E-12
115	4.9486E+02	5.8454E-10	9.6925E+00	1.1530E-11
120	1.4886E+02	3.7752E-10	3.2705E+01	4.6346E-11
125	7.4150E+01	3.1175E-10	1.0743E+01	2.5816E-11
130	3.9554E+01	2.4003E-10	3.4461E+00	1.1597E-11
135	6.7305E+01	2.4253E-10	1.2958E+00	2.1858E-11
140	6.2313E+01	1.2816E-10	4.1279E+00	1.4562E-11
145	4.6856E+03	8.8545E-09	7.3341E+02	1.1682E-09
150	4.8977E+02	1.0277E-09	2.3644E-01	5.5400E-14



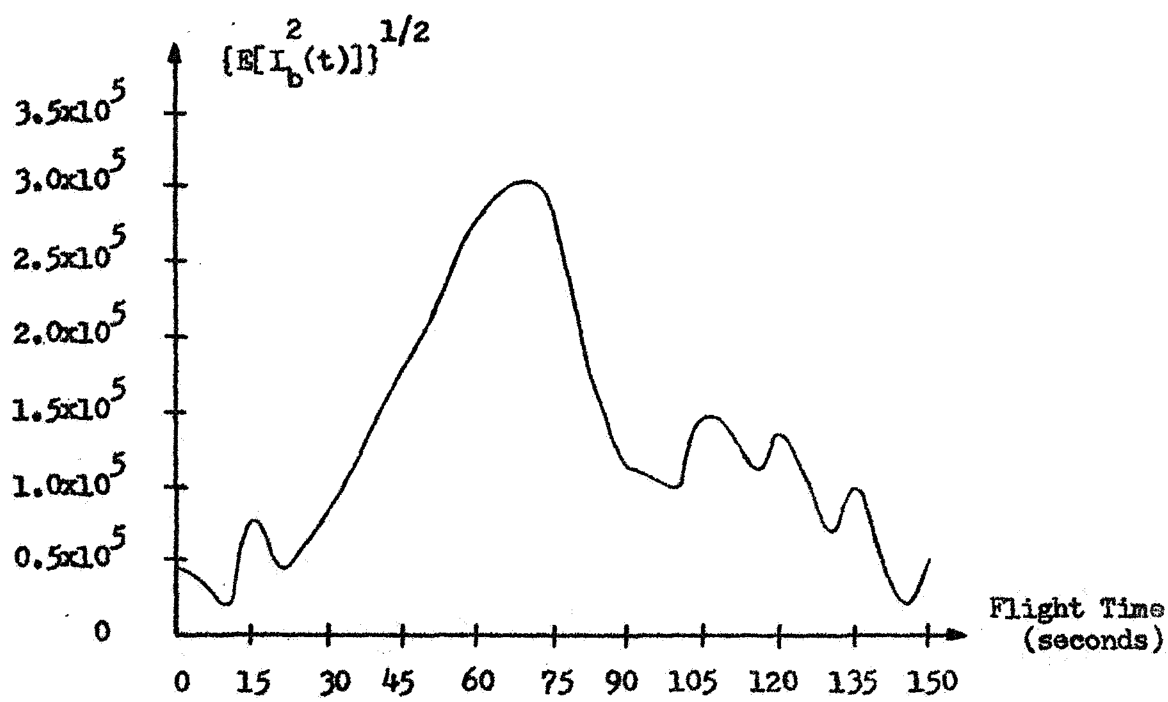


Figure 6.14 Standard Deviations of β and I_b

$$\sigma_y(150) = 147$$

$$\sigma_{\dot{y}}(150) = 0.998$$

$$\sigma_{\xi}(150) = 0.0197.$$

So the probability that these responses were outside their respective bounds of $\gamma_1 = 3000$, $\gamma_2 = 40$, and $\gamma_3 = 0.0873$ at the terminal time is also very small. Thus $K_5{}^*(t)$ was also a "good" one in producing small terminal responses. The above results give another indication that using PGM to minimize J_N is a useful technique for obtaining good load-relief controllers.

In Figure 6.15, the values of Δ_i computed in the PGM and DGIM iterations are plotted. Remember that Δ_i (defined in (6-13)) is a measure of how well the necessary conditions for equivalence, given in part 1) of Theorem 4.1, are being satisfied at the ith iteration. In interpreting Figure 6.15, it should be noted that the Δ_i computed for the PGM sequence is with respect to J_N, and the Δ_i computed for the DGIM sequence is with respect to J_s. It was found that Δ_i for the DGIM sequence changed very little, and thus little progress was made towards satisfying the equivalence conditions. For the PGM sequence, however, Δ_i did decrease rapidly. A stopping condition which would guarantee "approximate equivalence" (such as requiring $\Delta_i \leq 0.01$ in the first example in this chapter) was not used in this example. Instead, the iterations were continued until "good" controls (as measured by J_s) resulted.

Iteration Sequence 2

In this sequence, the values of γ_i used in J_s and of δ_i used in J_N

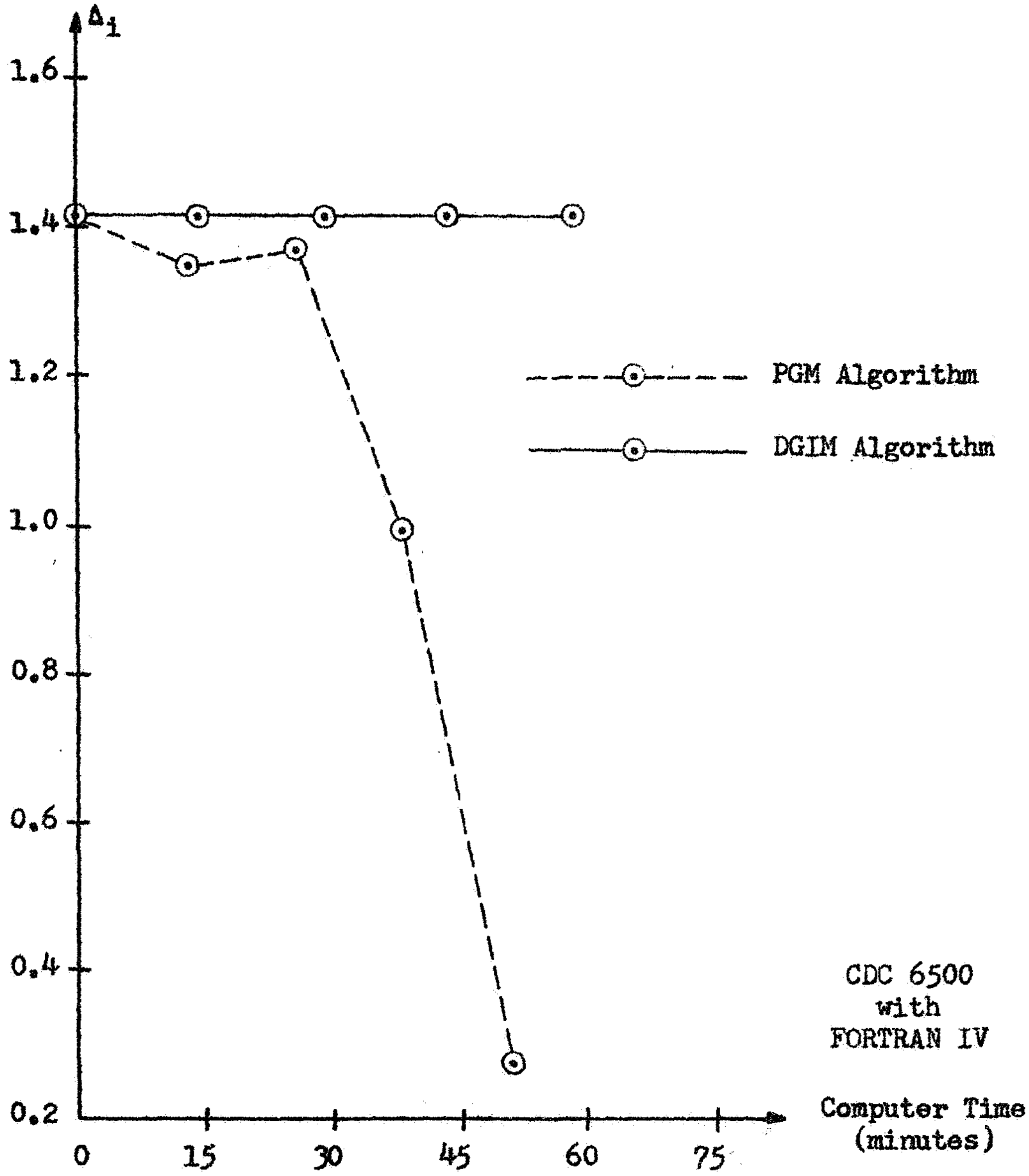


Figure 6.15 Δ_i in Load-Relief Problem, Sequence 1

are as follows:

$$\gamma_1 = 3000 \qquad \delta_1 = 3.0 \times 10^3$$

$$\gamma_2 = 40 \qquad \delta_2 = 0.4$$

$$\gamma_3 = 5.94 \times 10^{-2} \qquad \delta_3 = 1.0 \times 10^{-4}$$

$$\gamma_4 = 8.73 \times 10^{-2} \qquad \delta_4 = 1.0 \times 10^{-4}$$

$$\gamma_5 = 2.25 \times 10^6 \qquad \delta_5 = 1.0 \times 10^{10}$$

$$\delta_6 = 1.0 \times 10^{-5}$$

$$\delta_7 = 1.0 \times 10^{11}$$

The γ_i are similar to those in the first sequence, as are the δ_i. The value of λ used in DGIM was $\lambda = 0.9$. The values of the $Q_F(T)$ and $Q(t)$ quadratic coefficient matrices which were used to start the iteration sequence were suggested by Skelton in private correspondence. The form of the matrices is the same as that for the initial coefficient matrices in the first iteration sequence (see equations (6-43) and (6-44)). The nonzero elements of the initial $Q_F(T)$ matrix are:

$$Q_{F_{11}}(T) = 1.1111 \times 10^{-2}$$

$$Q_{F_{22}}(T) = 2.041 \times 10^{-3}$$

$$Q_{F_{33}}(T) = 1.42 \times 10^{7} .$$

The form of the initial $Q(t)$ was also the same as that in the first sequence, except that the values of Q_{ii}, $i = 4,5,6,7$, were constant over the whole time interval, and were given by:

$$Q_{44} = 7.8799 \times 10^{5}$$

$$Q_{55} = 1.2346 \times 10^{-10}$$

$$Q_{66} = 7.8799 \times 10^{5}$$

$$Q_{77} = 7.716 \times 10^{-12} .$$

The results of applying PGM to the problem of minimizing J_N in this sequence are shown in Figure 6.16. Again, the sequence of values is monotonically decreasing, but the percentage of change in J_N from the initial value is not very great. The evaluation of the $\{\hat{s}_i\}$ sequence obtained by PGM is shown in Figure 6.17. In this case, the sequence $\{J_s(\hat{s}_i)\}$ is also monotone decreasing. As in the first sequence, the DGIM algorithm was applied to minimizing J_s, and the result is also shown in Figure 6.17.

The decrease in the J_s performance index achieved by both algorithms is not very substantial, as can be seen in the figure. This was partially due to the fact that the initial value of $J_s = 3.755 \times 10^{-8}$ was very small as an upper bound to a probability. Thus, the initial feedback coefficient, $K_o(t)$, was a very good one, and perhaps not much improvement could be expected. Another reason could be that the λ used in DGIM and the δ_i's chosen for J_N may have been poorly selected. The problem of choosing λ in DGIM is one of the intrinsic defects in that algorithm; Skelton does not give detailed instructions as to the best way of making that choice. The choice of the δ_i's to be used in J_N is also a matter of judgement and trial-and-error, in trying to "match" the performance

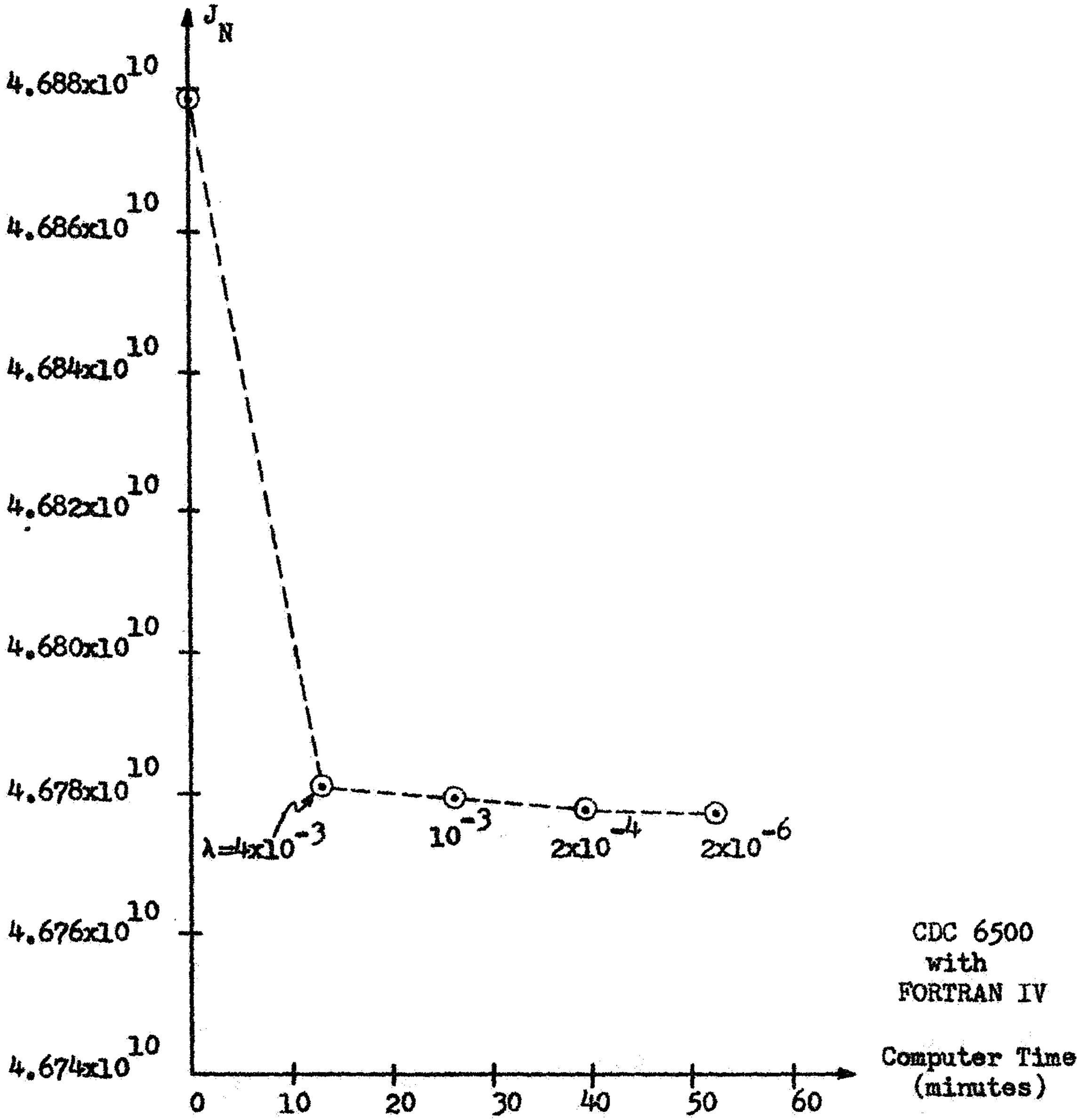


Figure 6.16 J_N in Load-Relief Problem, Sequence 2

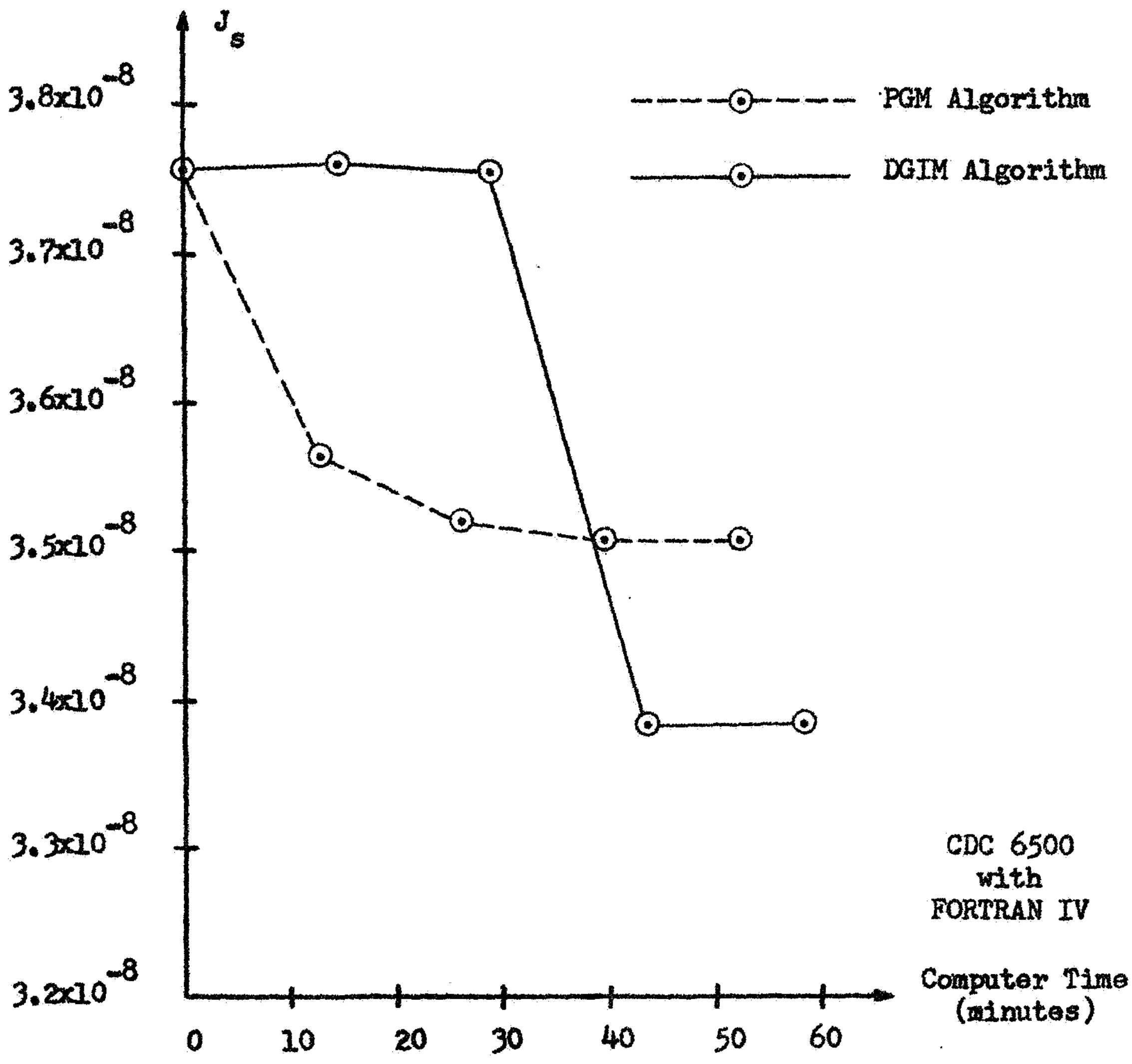


Figure 6.17 J_s in Load-Relief Problem, Sequence 2

indices J_N and J_s in some sense. Once the δ_i's are chosen, however, the PGM algorithm can be applied to minimizing J_N automatically; no engineering judgement or guesswork is necessary.

Let us now consider the overall results obtained in the two iteration sequences. It was shown that the PGM algorithm could be successfully applied to the problem of minimizing J_N, subject to side-conditions in the form of high-order differential equations. It was also verified that the DGIM algorithm could be successfully applied to the problem of minimizing Skelton's upper-bound performance index, subject to the same differential side-conditions. Skelton had, of course, demonstrated this earlier in [2.4] and [6.1]. A practical result was that, if the δ's in J_N were chosen judiciously, the controls generated by using PGM to minimize J_N were useful ones in Skelton's load-relief problem. The advantage of using this second, suboptimal method in a practical problem was that the PGM algorithm was an automatic one, and was known to converge if the hypotheses of Theorem 5.1 were satisfied.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Discussion of Research

The research discussed in the previous chapters was directed toward the solution of a type of stochastic optimal control problem (the "J-problem") posed in section 2.2. Skelton in [2.4] studied a specific case of the J-problem, in which the performance index was the probability-upper-bound one discussed in section 6.3.1 and Appendix F. He recognized that a well-known "quadratic" control problem (the "J_Q-problem" stated in section 2.3) had properties similar to his specific J-problem, and that the known solution to the J_Q-problem could be used in solving his problem. The main contribution of the research discussed here is the formulation of the J-problem as one of minimizing a nonlinear functional on a set in a Hilbert space. In this formulation, the J_Q-problem takes on a special significance, that of minimizing a linear functional on the same set in the space. Conditions were derived in Theorem 4.1, under which the nonlinear and linear functionals took on their minimum values at the same point in the set. When this occurred, the problems of minimizing the two functionals were said to be "equivalent." Skelton introduced this concept of equivalence of stochastic control problems in [2.4]; however, the function space approach discussed here gives a clearer, geometric interpretation of this concept.

A function space algorithm of Dem'yanov was applied to solving the general class of problems, and conditions under which the algorithm converged were derived in Theorem 5.1. This algorithm (the perturbed gradient method) involved solving a sequence of linear functional minimization problems to find the minimum of a nonlinear functional iteratively. The PGM algorithm, as well as Skelton's DGIM algorithm, was applied to the solution of two example problems. Both algorithms attained a given stopping condition in the first example (see section 6.2), which meant that numerical convergence was achieved. This also meant that the equivalence conditions in Theorem 4.1 were achieved numerically (i.e., within the desired computational accuracy). This was a significant step in the research, for the following reason. Skelton had used his DGIM in [2.4] and [6.1] to obtain "good" load-relief controllers, as measured by his probability-upper-bound performance index (see section 6.3.1). However, due to enormous consumption of computer time, he did not make any attempt to continue the operation of DGIM until the equivalence conditions were met (even numerically). Thus, the success obtained in achieving the stopping condition and minimizing the performance index in the first example showed that an equivalent J_Q-problem could be found and that the equivalence concept was a valid one. In the second example, the PGM algorithm was used in a suboptimal approach to solving a load-relief problem similar to the one studied by Skelton in [6.1]. This approach led to good controls, as measured by Skelton's "probability upper bound" performance index. Thus PGM and the supporting function-space approach were shown to be useful in solving a practical problem involving a high-order dynamic system model.

7.2 Suggestions for Future Investigation

The function-space formulation of the type of stochastic control problem discussed above provided a useful theoretical framework for the research recorded in this thesis. Within this framework, a number of important theoretical questions have not been answered and remain for future investigation. Some of these problems are as follows:

1) General conditions on the admissible control set and the dynamic equations, which would guarantee the convexity of the set α (see Definition 3.2), have not yet been found. An approach to determining these conditions was outlined in section 4.4 for a special case, but a general convexity proof is not yet available. Convexity of α is, of course, required in the derivation of equivalence conditions in Theorem 4.1, and is also required so that the PGM algorithm can be applied to the J-problem.

2) The question of the existence of a solution to the J-problem (i.e., whether a minimum value of the functional J on α exists) has not been answered. In the function space formulation, such an existence proof would require some type of continuity requirement on the J-functional, plus some type of compactness requirement on the set α. For example, if J is a continuous functional, defined on a set α which is compact in itself (i.e., every infinite subset of α contains a sequence which converges to a limit point in α), then a minimum point of J on α exists (see, e.g., [C.1], p. 35). Conditions on the J-problem which would guarantee that these requirements are met have not yet been found.

3) Assertion 2.1, concerning the known formal solution to the "quadratic" problem, has not yet been proven rigorously, as far as is

known. The solution to the "quadratic" problem is a key element in the equivalence concept and in the computational methods discussed in Chapter 5. Thus, Assertion 2.1 should be given further study, as new results in stochastic control theory become available.

4) The hypotheses in theorems 4.1 and 5.1 are very strong ones; perhaps the proofs of the theorems could be refined so that weaker hypotheses could be invoked. For example, local convexity and compactness conditions on α, plus other side conditions on J, might replace the first two hypotheses in Theorem 4.1.

In addition to the theoretical questions discussed above, a number of computational problems are still open to investigation:

5) The convergence properties of the DGIM algorithm (introduced by Skelton and discussed in section 5.2) have not been given sufficient study. The algorithm did satisfy the stopping condition when used in the first example in Chapter 6, and has been used by Skelton to obtain good controls. Thus, it seems possible that properties of J and α which would guarantee convergence of DGIM could be found.

6) More sophisticated procedures for finding the minimum of J on the "straight line" between $\hat{s}_i$ and $\hat{s}_i{}^*$ (in the PGM algorithm) could be investigated. The "walking" procedure used in the examples and described in Appendix G was relatively crude, but effective. Further studies of this "one-dimensional" minimization problem in function space should be performed, especially concerning the trade-offs to be made between computational complexity and speed of convergence of the PGM algorithm.

7) A number of algorithms for minimizing a function on a set in Euclidean space were described by Dem'yanov in [5.3]. The possibility

that some of these algorithms could be adapted to the function space and used in solving the J-problem should be investigated.

8) As discussed in section 6.3.2, the PGM algorithm could not be directly applied to the problem of minimizing J_s (the load-relief problem), due to difficulties in the computation of a solution to the Riccati equation (2-23). These difficulties should be investigated further. In particular, a good interpretation of a "quadratic" problem in which the coefficient Q(t) is identically (or nearly) zero over a finite time period is needed. A solution to this type of problem must be found if the PGM algorithm is to be applied directly to the load-relief problem.

The research described in this thesis raises a few other questions:

9) The disturbance noise and measurement noise in the stochastic problems considered were all assumed to be zero-mean. That is, the problems considered were all "perturbation" ones, in which deviations from some nominal trajectory were to be minimized. Thus, the investigation of a more general stochastic problem which involves non-zero-mean noise and non-zero initial conditions is a possible topic for future research. Also, the cases of "colored" disturbance and measurement noise, and of measurements which contain no noise (i.e., $N_w(t)$ is allowed to be singular) should be investigated.

10) In section 3.2, it was shown that the stochastic control problems defined in Chapter 2 could be reformulated as deterministic ones. Using this formulation, it is possible that some of the results in deterministic control theory (such as the maximum principle or dynamic programming) could be brought to bear on the J-problem. This

approach would not require a function space formulation, and would not make use of the known formal solution to the "quadratic" problem. It is a valid approach, however, and could be investigated further.

11) The idea of using the known solution to a particular problem in solving a more general class of problems was found to be a powerful one in the research described above. It led to the concept of "equivalence" of stochastic problems, and to a number of algorithms in which the known solution was a vital part of the iteration procedure. The application of this idea to other classes of control problems may be a fruitful approach, and should be investigated.

LIST OF REFERENCES

[1.1] Florentin, J. J., "Optimal Control of Continuous Time, Markov, Stochastic Systems," J. Electronics Control, vol. 10, 1961, pp. 473-488.

[1.2] Orford, R. J., "Optimal Stochastic Control Systems," J. Math. Analysis and Appl., vol. 6, 1963, pp. 419-429.

[1.3] Kounias, E., "Optimal Bounded Control with Linear Stochastic Equations and Quadratic Cost," J. Math. Analysis and Appl, vol. 16, 1966, pp. 510-537.

[1.4] Kushner, H. J., "Some Problems and Some Recent Results in Stochastic Control," IEEE International Convention Record, pt. 6, 1965, pp. 108-116.

[1.5] Kushner, H. J., "On the Status of Optimal Control and Stability for Stochastic Systems," IEEE International Convention Record, vol. 14, pt. 6, 1966, pp. 143-151.

[1.6] Paiewonsky, B., "Optimal Control: A Review of Theory and Practice," AIAA Journal, vol. 3, no. 11, Nov., 1965, pp. 1985-2006.

[1.7] Mayne, D. Q., "Recent Approaches to the Control of Stochastic Processes," J. Inst. Maths. Applications, vol. 3, 1967, pp. 46-59.

[2.1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," J. Basic Engineering (ASME), March, 1960, pp. 35-45.

[2.2] Bryson, A. E., and Johansen, D. E., "Linear Filtering for Time-Varying Systems Using Measurements Containing Colored Noise," IEEE Trans. Automatic Control, vol. AC-10, no. 1, Jan. 1965, pp. 4-10.

[2.3] Kalman, R. E., and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory, "J. Basic Engineering (ASME), March, 1961 pp. 95-108.

[2.4] Skelton, G. B., "Launch Booster Gust Alleviation," paper no. 66-969, AIAA Third Annual Meeting, 1966.

[2.5] Florentin, J. J., "Partial Observability and Optimal Control," J. Electronics and Control. vol. 13, 1962, pp. 263-279.

[2.6] Tou, J. S., Optimum Design of Digital Control Systems, Academic Press, N. Y., 1963.

[2.7] Wonham, W. M., "Stochastic Problems in Optimal Control," IEEE International Convention Record, Part 2, 1963, pp. 114-124.

[2.8] Wonham, W. M., Lecture Notes on Stochastic Control, Brown University Lecture Notes 67-2, February, 1967.

[3.1] Jazwinski, A. H., "On Optimal Stochastic Midcourse Guidance," Proc. JACC, 1967, pp.317-325.

[3.2] Kushner, H. J., "On the Differential Equations Satisfied by Conditional Probability Densities of Markov Processes, with Applications," J. SIAM Control, vol. 2, no. 1, 1964, pp. 106-119.

[3.3] Mortensen, R. E., "A Priori Open Loop Optimal Control of Continuous Time Stochastic Systems," International J. of Control, vol. 3, no. 2, 1966, pp. 113-127.

[3.4] Dunford, N., and Schwartz, J. T., Linear Operators, Interscience, London, 1958.

[3.5] Rudin, W., Real and Complex Analysis, McGraw-Hill, New York, 1966.

[3.6] Vainberg, M. M., Variational Methods for the Study of Nonlinear Operators, Holden-Day, San Francisco, 1964.

[3.7] Kantorovich, L. V., "Functional Analysis and Applied Mathematics," Uspechi Mat. Nauk., vol. 3, 1948, pp. 89-185; English translation, Nat. Bureau of Standards, 1953.

[3.8] Goldstein, A. A., "Minimizing Functionals on Hilbert Space," From Computing Methods in Optimization Problems, ed. by A. V. Balakrishnan and L. W. Neustadt, Academic Press, New York, 1964, pp. 159-165.

[4.1] Dem'yanov, V. F., and Rubinov, A. M., "The Minimization of a Smooth Convex Functional on a Convex Set," J. SIAM Control, vol. 5, no. 2, 1967, pp. 280-294.

[4.2] Kaplan, W., Advanced Calculus, Addison-Wesley, Reading, Mass., 1952.

[4.3] Gnekendo, B. V., The Theory of Probability, Chelsea, New York, 1962.

[4.4] Coddington, E., and Levinson, N., Theory of Ordinary Differential Equations, McGraw-Hill, New York, 1955.

[5.1] Blum, E. K., "Minimization of Functionals with Equality Constraints," J. SIAM Control, vol. 3, no. 2, 1965, pp. 299-316.

[5.2] Balakrishnan, A. V., "An Operator Theoretic Formulation of a Class of Control Problems and a Steepest Descent Method of Solution," J. SIAM Control, vol. 1, no. 2, 1963, pp. 109-127.

[5.3] Dem'yanov, V. F., "Minimization of Functions on Convex Bounded Sets," Cybernetics, vol. 1, no. 6, 1965, pp. 77-87 (English translation of Kibernetika).

[5.4] Gilbert, E. G., "An Iterative Procedure for Computing the Minimum of a Quadratic Form on a Convex Set," J. SIAM Control, vol. 4, no. 1, 1966, pp. 61-80.

[6.1] Edinger, L. D., Skelton, G. B., et. al., Design of a Load-Relief Control System, NASA Report no. NASA-CR-61169, Accession no. N67-27304, May 9, 1966.

[B.1] Kalman, R. E., "New Methods and Results in Linear Prediction and Estimation Theory," Tech. Rept. 61-1, Research Institute for Advanced Studies, Baltimore, Maryland, 1961.

[B.2] Kushner, H. J., "Optimal Stochastic Control," IRE Trans. Automatic Control, Oct. 1962, pp. 120-122.

[B.3] Kushner, H. J., "On the Dynamical Equations of Conditional Probability Density Functions, with Applications to Optimal Stochastic Control Theory," J. Math. Analysis and Applic., vol. 8, 1964, pp. 332-344.

[C.1] Liusternik, L. A., and Sobolev, V. J., Elements of Functional Analysis, Ungar, New York, 1961.

[C.2] Vulikh, B. Z., Introduction to Functional Analysis, Addison-Wesley, Reading, Mass., 1963.

[D.1] Kantorovich, L. V., and Akilov, G. P., Functional Analysis in Normed Spaces, Macmillan, New York, 1964.

[D.2] Rudin, W., Principles of Mathematical Analysis, McGraw-Hill, New York, 1964.

[F.1] Bendat, J. S., Principles and Applications of Random Noise Theory, Wiley, 1958, Chapters 3 and 10.

APPENDIX A

ANALYTIC APPROACH TO EQUIVALENCE (SKELTON)

In his paper on wind-gust effects on launch boosters [2.4], Skelton derived necessary conditions for two stochastic problems to be equivalent, in the sense described in Section 3.5. The derivation is reproduced here to show the analytic method used and to complete the discussion of equivalence.

In this approach, the J-problem and J_Q-problem are defined as in Chapter 3, except that the set of admissible controls is:

$$U_L = \left\{ \begin{array}{l} u:u \text{ is a linear function of the} \\ \text{measurements } z(\tau),\ \tau \in [t_o, t) \end{array} \right\} , \qquad \text{(A-1)}$$

and J is the upper bound index given by (6-34) through (6-39).

For notational convenience, an admissible control u will be written in the following form:

$$u = L(t,\ z(\tau),\ \tau \in [t_o, t)) = L(t,\ z) \quad . \qquad \text{(A-2)}$$

Now, assume that a solution to the J-problem exists, and is given by

$$u^* = L_o(t,\ z) \quad . \qquad \text{(A-3)}$$

That is, J is minimized over all admissible controls by L_o.

Now, consider a perturbation on L_o:

$$L(t,\ z) = L_o(t,\ z) + \varepsilon\ L_1(t,\ z) \ , \qquad \text{(A-4)}$$

where ϵ is a "small" real number, and L_1 is a linear function of t and z. Then L(t, z) is an admissible control in U_L. Then it can be shown that the response covariance matrix S(t) (defined in (2-17)) can be written as a polynomial in ϵ:

$$S(t) = S_o(t) + \epsilon\, S_1(t) + \epsilon^2 S_2(t) + \ldots + \epsilon^n S_n(t, \epsilon)\ . \tag{A-5}$$

As mentioned by Skelton in [2.4], $S_o(t)$ is the response covariance matrix which results if $u = L_o(t, z)$; the matrices $S_1, S_2, \ldots S_{n-1}$ are dependent on L_o and L_1 but are independent of ϵ. The last matrix S_n is dependent on ϵ, however.

Now, consider the deterministic form of the performance indexes J and J_Q, as defined in (3-16) and (3-17), respectively. Each index is a function of S(t). Thus, if the appropriate derivatives of $f_1[S(T)]$ and $f_2[S(t)]$ in J exist and are continuous, J and J_Q can be written as polynomials in ϵ:

$$J = J^o + \epsilon\, J^1 + \epsilon^2\, J^2 + \epsilon^3\, J^3\,(\epsilon) \tag{A-6}$$

$$J_Q = J_Q^o + \epsilon\, J_Q^1 + \epsilon^2\, J_Q^2 + \epsilon^3\, J_Q^3\,(\epsilon)\ , \tag{A-7}$$

where J^o and J_Q^o are the values of J and J_Q, respectively, using $u = L_o(t, z)$. Then, as described by Skelton, J^1, J^2, J_Q^1, and J_Q^2 are functions of L_o and L_1, but are independent of ϵ. The "third variations" J^3 and J_Q^3 are dependent on L_o, L_1, and ϵ.

Now, assume that a J_Q-problem that is equivalent to the given J-problem exists and is specified by the coefficient matrices $Q_F(T)$ and Q(t). That is, $L_o(t, z)$ minimizes both J and J_Q. Then it is clear that

$$J^1 = J_Q^1 = 0 , \tag{A-8}$$

since a necessary condition for minimization of J and J_Q is that the "first variation" of each equals zero. The equality of J^1 and J_Q^1 is then the required necessary condition for the equivalence of the J- and J_Q-problems. These "first variations" can be written in the form

$$J^1 = \mathrm{Tr}\left\{ \left[\frac{\partial f_1}{\partial S}\bigg|_{S_o(T)}\right] S_1(T) + \int_{t_o}^{T} \left[\frac{\partial f_2}{\partial S(t)}\bigg|_{S_o(t)}\right] S_1(t)\, dt \right\} \tag{A-9}$$

$$J_Q^1 = \mathrm{Tr}\left\{Q_F(T)\, S_1(T) + \int_{t_o}^{T} Q(t)\, S_1(t)\, dt\right\} . \tag{A-10}$$

Clearly, $J^1 = J_Q^1$ if

$$Q_F(T) = \frac{\partial f_1}{\partial S}\bigg|_{S_o(T)} \quad \text{and} \quad Q(t) = \frac{\partial f_2}{\partial S(t)}\bigg|_{S_o(t)} . \tag{A-11}$$

These are the necessary conditions for the equivalence of the J- and J_Q-problems. These same conditions are derived using the geometric approach in Chapter 4; additional conditions are placed on J to insure that these conditions are also sufficient.

APPENDIX B

DERIVATION OF RESPONSE COVARIANCE MATRIX

Once $K(t)$ is specified, the response covariance matrix $S(t)$ is completely defined by the stochastic system equations (2-1) to (2-4), the Kalman filter equations (2-10) to (2-13), and the error covariance equations (2-14) and (2-15). For convenience, however, an explicit expression for $S(t)$ in terms of the noise parameters and $K(t)$ is derived in this appendix. The Kalman filter terminology and results are assumed in this derivation.

From (2-17), we have

$$S(t) = E[r(t)r'(t)] \; . \tag{B-1}$$

If we define

$$F(t) = C(t) - D(t)K(t) \; , \tag{B-2}$$

and use (2-13), (2-9), and (B-2) in (2-3) we have

$$r(t) = F(t)x(t) + D(t)K(t)\tilde{x}(t|t) \; , \tag{B-3}$$

where $\tilde{x}(t|t)$ is defined in (2-13).

So

$$\begin{aligned} S(t) = E\{&[F(t)x(t) + D(t)K(t)\tilde{x}(t|t)] \\ &[x'(t)F'(t) + \tilde{x}'(t|t)K'(t)D'(t)]\} \; . \end{aligned} \tag{B-4}$$

Now, define

$$C_x(t) = E[x(t)x'(t)] \ . \tag{B-5}$$

Also,

$$E[\tilde{x}(t|t)x'(t)] = E[x(t)\tilde{x}'(t|t)] = E_k(t) \ , \tag{B-6}$$

since $\hat{x}$ (the Kalman filter state estimate) and $\tilde{x}$ are independent (see Kalman [B.1], p. 32). The matrix $E_k(t)$ is defined in (2-12).

Then (B-4) can be written

$$\begin{aligned} S(t) = {} & [C(t) - D(t)K(t)]C_x(t)[C'(t) - K'(t)D'(t)] \\ & + D(t)K(t)E_k(t)C'(t) + C(t)E_k(t)K'(t)D'(t) \\ & - D(t)K(t)E_k(t)K'(t)D'(t) \ , \end{aligned} \tag{B-7}$$

which is the desired expression for $S(t)$. Now, the differential equation which $C_x(t)$ satisfies will be derived.

In this derivation, the finite-difference representation of the system equations will be used instead of the representation in (2-1):

$$\begin{aligned} \Delta x(t) &= x(t + \Delta t) - x(t) \\ &= A(t)x(t)\Delta t + B(t)u\,\Delta t + \Delta v_t \ , \end{aligned} \tag{B-8}$$

where $\Delta v_t = v(t + \Delta t) - v(t)$, and $v(t)$ is a Wiener process with independent increments such that

$$E[\Delta v_t] = 0 \ , \tag{B-9}$$

and
$$E[\Delta v_t \Delta v'_{t+k\Delta t}] = \begin{cases} 0 \text{ if } k = 1,2,\ldots \\ N_v(t)\Delta t \text{ if } k = 0 \,, \end{cases} \tag{B-10}$$

for all $\Delta t > 0$, $t \in [t_o, T)$, $(t + k\Delta t) \in [t_o, T]$.

The representation in (B-8) to (B-10) then becomes completely equivalent to the representation in (2-1) to (2-8) as $\Delta t \to 0$ (see, e.g. Kushner [B.2], [B.3], [3.2]).

Using (2-9) in (B-8), we have

$$\begin{aligned} x(t + \Delta t) = x(t) &+ A(t)x(t)\Delta t \\ &- B(t)K(t)\hat{x}(t|t)\Delta t + \Delta v_t \,. \end{aligned} \tag{B-11}$$

Now, form

$$C_x(t + \Delta t) = E[x(t + \Delta t)x'(t + \Delta t)] \,. \tag{B-12}$$

Using (B-11) and (B-9) in (B-12), and noting that

$$E[x(t)\Delta v'_t] = 0 = E[\hat{x}(t)\Delta v'_t] \,, \tag{B-13}$$

we have

$$\begin{aligned} C_x(t + \Delta t) - C_x(t) = C_x(t)A'(t)\Delta t &+ A(t)C_x(t)\Delta t \\ &- E[x(t)\hat{x}'(t|t)]K'(t)B'(t)\Delta t \\ &- B(t)K(t)E[\hat{x}(t|t)x'(t)]\Delta t \\ &+ N_v(t)\Delta t + o(\Delta t) \,, \end{aligned} \tag{B-14}$$

where $$\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0 \,. \tag{B-15}$$

Now, note that

$$E[x(t)\hat{x}'(t|t)] = E[\hat{x}(t|t)x'(t)] = C_x(t) - E_k(t). \qquad (B\text{-}16)$$

Using (B-16) in (B-14), dividing both sides of (B-14) by Δt, and taking the limit as $\Delta t \to 0$, we have:

$$\frac{dC_x(t)}{dt} = [A(t) - B(t)K(t)]C_x(t) + C_x(t)[A'(t) - K'(t)B'(t)]$$
$$+ B(t)K(t)E_k(t) + E_k(t)K'(t)B'(t) + N_v(t) \, , \qquad (B\text{-}17)$$

where $\quad C_x(t_o) = 0 \, . \qquad (B\text{-}18)$

Equations (B-7), (B-17), and (B-18) thus define $S(t)$.

APPENDIX C

CONSTRUCTION OF THE HILBERT SPACE σ

In this Appendix it is shown that the space σ, defined in Section 3.3, is a Hilbert space. Certain results from Dunford and Schwartz [3.4] will be used.

In [3.4], p. 255, the following definition is given:

<u>Definition C.1.</u> The <u>direct sum</u>

$$X = X^1 \oplus X^2 \oplus \ldots \oplus X^n \tag{C-1}$$

of the vector spaces $X^1, X^2, \ldots X^n$ is defined to be the product space of the X^i's, in which addition and scalar multiplication are defined by:

$$\begin{aligned} x + y &= [x^1 \;\; x^2 \;\ldots\; x^n] + [y^1 \;\; y^2 \;\ldots\; y^n] \\ &= [(x^1 + y^1)\;(x^2 + y^2)\;\ldots\;(x^n + y^n)] \;, \end{aligned} \tag{C-2}$$

$$\begin{aligned} \alpha x &= \alpha[x^1 \;\; x^2 \;\ldots\; x^n] \\ &= [\alpha x^1 \;\; \alpha x^2 \;\ldots\; \alpha x^n] \;, \end{aligned} \tag{C-3}$$

where $x, y \in X$; $x^i, y^i \in X^i$, $i = 1,2,\ldots,n$, and α is a real scalar.

If the X^i's are Hilbert spaces, the following holds ([3.4], p.256):

<u>Definition C.2.</u> For each $i = 1,2,\ldots,n$, let X^i be a Hilbert space in which the inner product $(.,.)_i$ is defined. The <u>direct sum of the Hilbert spaces</u> $X^1, X^2,\ldots X^n$ is the linear space

$$X = X^1 \oplus X^2 \oplus \dots \oplus X^n ,$$

in which the inner product is defined:

$$\begin{aligned}(x,y) &= ([x^1 \;\; x^2 \dots x^n], [y^1 \;\; y^2 \dots y^n]) \\ &= \sum_{i=1}^{n} (x^i, y^i)_i \end{aligned} \qquad \text{(C-4)}$$

where $x, y \in X$; $x^i, y^i \in X^i$, $i = 1,2,\dots n$.

The main result to be used is ([3.4], p. 257):

Lemma C.1

If $\{X^i\}$, $i = 1,2,\dots n$ is a family of Hilbert spaces, their direct sum is a Hilbert space.

To show that σ is a Hilbert space, let X^1 in the above Lemma be an L^2-space, whose elements are of the form $e_1(t)$, $t \in [t_o, T]$, where e_1 is a measurable real scalar function on its domain. The inner product in X^1 is defined as:

$$(e_1(t), \bar{e}_1(t))_1 = \int_{t_o}^{T} e_1(t)\, \bar{e}_1(t) dt, \qquad \text{(C-5)}$$

where $e_1(t)$ and $\bar{e}_1(t)$ are both in X^1. Similarly, let $X^2,\dots X^k$ also be L^2-spaces, with elements of the form $e_1(t)$, $e_2(t),\dots e_k(t)$, real scalar functions defined on $[t_o, T]$. Also, let the inner products in X^2, $X^3,\dots X^k$ be defined as in (C-5). Since an L^2-space is a Hilbert space (see [C.1], p. 74), each X^i, $i = 1,2,\dots k$ is a Hilbert space. Let X^{k+1} be E^k, k-dimensional Euclidean space, on which the usual scalar product is defined. Then X^{k+1} is also a Hilbert space (see, e.g., Vulikh [C.2], p. 155).

Now, identify $e_1(t)$, $e_2(t),\ldots,e_k(t)$ as the k components of the vector $e(t)$ in (3-6); and let e_F be the typical element in X^{k+1}. Then, by definitions C.1 and C.2, and by Definition 3.1 of σ, it can be shown that σ is precisely the direct sum of the X^i, $i = 1,2,\ldots k+1$. So by Lemma C.1, σ is a Hilbert space.

APPENDIX D

DIFFERENTIALS AND GRADIENT VECTOR OF $J(\hat{s})$

In this Appendix, Theorem 3.1, which gives explicit expressions for the first and second Gateaux differentials and the gradient vector of $J(\hat{s})$, is proved. The following definitions will be used in the proof:

Definition D.1 (from [3.6], p.35): If, at $\hat{s} \in \sigma$, and for all $\hat{e} \in \sigma$,

$$\lim_{\gamma \to 0} \frac{J(\hat{s} + \gamma\hat{e}) - J(\hat{s})}{\gamma} = \nabla J(\hat{s}, \hat{e}) \tag{D-1}$$

exists, then $\nabla J(\hat{s}, \hat{e})$ is called the Gateaux differential (or weak differential) of the functional J at the point $\hat{s}$, in the direction $\hat{e}$. Further, from [C.1], p.184, an equivalent definition is:

$$\nabla J(\hat{s}, \hat{e}) = \frac{d}{d\gamma} J(\hat{s} + \gamma\hat{e})\Big|_{\gamma = 0} . \tag{D-2}$$

Definition D.2 (from [D.1], p.675): If, at $\hat{s} \in \sigma$, and for all $\hat{e}, \hat{\eta} \in \sigma$,

$$\lim_{\gamma \to 0} \frac{\nabla J(\hat{s} + \gamma\hat{e}, \hat{\eta}) - \nabla J(\hat{s}, \hat{\eta})}{\gamma} = \nabla^2 J(\hat{s}, \hat{e}, \hat{\eta}) \tag{D-3}$$

exists, then $\nabla^2 J(\hat{s}, \hat{e}, \hat{\eta})$ is called the second Gateaux differential of J at the point $\hat{s}$, for increments $\hat{e}$ and $\hat{\eta}$. From [C.1], p.189, an equivalent definition is:

$$\nabla^2 J(\hat{s}, \hat{e}, \hat{\eta}) = \frac{d}{d\gamma} \nabla J(\hat{s} + \gamma\hat{e}, \hat{\eta})\Big|_{\gamma = 0} \tag{D-4}$$

The several parts of Theorem 3.1 are then proved as follows:

1) Using the definition of J in (3-16), and remembering that $\hat{s} = [s(T), s(t)]$, $\hat{e} = [e_F, e(t)]$, we have:

$$J[\hat{s} + \gamma\hat{e}] = f_1[s(T) + \gamma e_F] + \int_{t_o}^{T} f_2[s(t) + \gamma e(t)]dt \ . \tag{D-5}$$

Use (D-5) in (D-2) of Definition D.1:

$$\nabla J(\hat{s}, \hat{e}) = \frac{d}{d\gamma} f_1[s(T) + \gamma e_F]\Big|_{\gamma = 0} + \int_{t_o}^{T} \frac{d}{d\gamma} f_2[s(t) + \gamma e(t)]\Big|_{\gamma = 0} dt \tag{D-6}$$

But now

$$\frac{d}{d\gamma} f_1[s(T) + \gamma e_F]\Big|_{\gamma = 0} = \frac{\partial f_1}{\partial s}[\xi(\gamma)] \cdot \frac{d\xi(\gamma)}{d\gamma}\Big|_{\gamma = 0} , \tag{D-7}$$

where $\quad \xi(\gamma) = s(T) + \gamma e_F \ ,$ (D-8)

and the dot indicates the Euclidean inner product. Carrying the indicated operations in (D-7) through, we have:

$$\frac{d}{d\gamma} f_1[s(T) + \gamma e_F]\Big|_{\gamma = 0} = \frac{\partial f_1}{\partial s}[s(T)] \cdot e_F \ , \tag{D-9}$$

where $\frac{\partial f_1}{\partial s}$ is defined in (3-26).

Similarly,

$$\frac{d}{d\gamma} f_2[s(t) + \gamma e(t)]\Big|_{\gamma = 0} = \frac{\partial f_2}{\partial s}[s(t)] \cdot e(t) \qquad \text{(D-10)}$$

for every $t \in [t_o, T]$, and where $\frac{\partial f_2}{\partial s}$ is also defined in (3-26). Note that the above partial derivative vectors exist by hypothesis.

Now, define the vector:

$$DJ(\hat{s}) = \left[\frac{\partial f_1}{\partial s}, \frac{\partial f_2}{\partial s}(t)\right]\Big|_{\hat{s}} . \qquad \text{(D-11)}$$

To show that $DJ(\hat{s}) \in \sigma$ for every $\hat{s} \in \sigma$, first note that a continuous function of a measurable function is measurable (see [D.2], p.238). If $\hat{s} \in \sigma$, $s(t)$ is Lebesgue measurable; since $\frac{\partial f_2}{\partial s}$ is continuous in s, $\frac{\partial f_2}{\partial s}$ is measurable and $\|DJ(\hat{s})\|\sigma$ is well defined. Since $\frac{\partial f_1}{\partial s}$ and $\frac{\partial f_2}{\partial s}$ are assumed to be finite for every $\hat{s} \in \sigma$, $\|DJ(\hat{s})\|\sigma$ is finite. Thus $DJ(\hat{s}) \in \sigma$ by Definition 3.1.

Since $\frac{\partial f_2}{\partial s}$ is measurable, the integral in (D-6) is defined. Using (D-9) and (D-10) in (D-6), we have:

$$VJ(\hat{s},\hat{e}) = \frac{\partial f_1[s(T)]}{\partial s} \cdot e_F + \int_{t_o}^{T} \frac{\partial f_2[s(t)]}{\partial s} \cdot e(t)dt \; . \qquad \text{(D-12)}$$

Then, by (D-11) and the inner product definition (3-10),

$$VJ(\hat{s}, \hat{e}) = (DJ(\hat{s}), \hat{e}) , \qquad \text{(D-13)}$$

which proves part 1 of the Theorem.

2) Using $\hat{\eta} = [\eta_F, \eta(t)]$ in (D-12) results in:

$$VJ(\hat{s} + \gamma\hat{e}, \hat{\eta}) = \frac{\partial f_1}{\partial s}[s(T) + \gamma e_F] \cdot \eta_F + \int_{t_o}^{T} \frac{\partial f_2}{\partial s}[s(t) + \gamma e(t)] \cdot \eta(t)dt . \qquad \text{(D-14)}$$

Then, by (D-4) in Definition D.2,

$$
\begin{aligned}
\nabla^2 J(\hat{s}, \hat{e}, \hat{\eta}) &= \frac{d}{d\gamma} \frac{\partial f_1}{\partial s}[s(T) + \gamma e_F] \cdot \eta_F \Big|_{\gamma = 0} \\
&\quad + \int_{t_0}^{T} \frac{d}{d\gamma} \frac{\partial f_2}{\partial s}[s(t) + \gamma e(t)] \cdot \eta(t) \Big|_{\gamma = 0} dt \, .
\end{aligned} \tag{D-15}
$$

Let

$$
\frac{\partial f_1}{\partial s} = \begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_k \end{bmatrix} , \qquad \eta_F = \begin{bmatrix} \eta_{F_1} \\ \eta_{F_2} \\ \cdot \\ \cdot \\ \cdot \\ \eta_{F_k} \end{bmatrix} . \tag{D-16}
$$

Then the first term on the right side of (D-15) becomes:

$$
\begin{aligned}
\text{TERM 1} &= \frac{d}{d\gamma} \frac{\partial f_1}{\partial s}[s(T) + \gamma e_F] \cdot \eta_F \Big|_{\gamma = 0} \\
&= \sum_{i=1}^{k} \frac{dF_i[s(T) + \gamma e_F]}{d\gamma} \eta_{F_i} \Big|_{\gamma = 0}
\end{aligned} \tag{D-17}
$$

As was done in (D-7), we can write

$$
\text{TERM 1} = \frac{\partial F_i}{\partial s}[\xi(\gamma)] \cdot \frac{d\xi(\gamma)}{d\gamma} \Bigg|_{\gamma = 0} , \tag{D-18}
$$

where $\xi(\gamma)$ is defined in (D-8), and $\frac{\partial F_i}{\partial s}$ is defined as was $\frac{\partial f_1}{\partial s}$. Carrying the operations in (D-18) through results in

$$
\text{TERM 1} = \frac{\partial F_i}{\partial s}[s(T)] \cdot e_F \tag{D-19}
$$

Then, letting e_{F_j} be the jth component of e_F, and using the definition of $\frac{\partial F_i}{\partial s}$, we have:

$$\frac{\partial F_i}{\partial s}[s(T)] \cdot e_F = \sum_{j=1}^{k} \frac{\partial^2 f_1}{\partial s_i \partial s_j} e_{F_j} . \tag{D-20}$$

Combine (D-17), (D-19), and (D-20):

$$\text{TERM 1} = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{\partial^2 f_1}{\partial s_i \partial s_j} \eta_{F_i} e_{F_j} . \tag{D-21}$$

Using the definition of $\frac{\partial^2 f_1}{\partial s^2}$ in (3-26), (D-21) becomes:

$$\frac{d}{d\gamma} \frac{\partial f_1}{\partial s}[s(T) + \gamma e_F] \cdot \eta_F \Big|_{\gamma = 0} = \frac{\partial^2 f_1[s(T)]}{\partial s^2} e_F \cdot \eta_F \tag{D-22}$$

Similarly, it can be shown that

$$\frac{d}{d\gamma} \frac{\partial f_2}{\partial s}[s(t) + \gamma e(t)] \cdot \eta(t) \Big|_{\gamma = 0} = \frac{\partial^2 f_2[s(t)]}{\partial s^2} e(t) \cdot \eta(t) \tag{D-23}$$

for every $t \in [t_o, T]$.

Define the vector:

$$D^2 J(\hat{s}, \hat{e}) = \left[\frac{\partial^2 f_1}{\partial s^2} e_F, \frac{\partial^2 f_2}{\partial s^2} e(t)\right]\Big|_{\hat{s}} \tag{D-24}$$

It can be shown that $\frac{\partial^2 f_2}{\partial s^2} e(t)$ is measurable by the same argument used in the case of $\frac{\partial f_2}{\partial s}(t)$ in the first part of the proof. So $D^2 J(\hat{s}, \hat{e}) \in \sigma$ by Definition 3.1. Using the inner product definition (3-10) in conjunction with (D-15) and (D-22) to (D-24), it follows that

$$\nabla^2 J(\hat{s}, \hat{s}, \hat{\eta}) = (D^2 J(\hat{s}, \hat{s}), \hat{\eta}) \; . \qquad \text{(D-25)}$$

and part 2 of the Theorem is proved.

3) By assumption, $DJ(\hat{s})$ is continuous in $\hat{s}$ in the σ-norm. That is, $\|DJ(\hat{s}_1) - DJ(\hat{s}_2)\|\sigma \to 0$ as $\|\hat{s}_1 - \hat{s}_2\|\sigma \to 0$. To show that $\nabla J(\hat{s}, \hat{s})$ is continuous in $\hat{s}$, use the Schwarz inequality:

$$|\nabla J(\hat{s}_1,\hat{s}) - \nabla J(\hat{s}_2,\hat{s})| = |(DJ(\hat{s}_1) - DJ(\hat{s}_2), \hat{s})| \qquad \text{(D-26)}$$

$$\leq \|DJ(\hat{s}_1) - DJ(\hat{s}_2)\|\sigma \quad \|\hat{s}\|\sigma \; .$$

Then, by the assumed continuity of $DJ(\hat{s})$ in $\hat{s}$, it can be seen that $|\nabla J(\hat{s}_1,\hat{s}) - \nabla J(\hat{s}_2,\hat{s})|$ goes to zero as $\|\hat{s}_1 - \hat{s}_2\|\sigma \to 0$. (By definition, if $\hat{s} \in \sigma$, $\|\hat{s}\|\sigma$ is finite). So ∇J is continuous in $\hat{s}$. The continuity of $\nabla^2 J(\hat{s}, \hat{s}, \hat{\eta})$ in $\hat{s}$ can be shown in exactly the same way, and so the proof of Theorem 3.1 is complete. Q.E.D.

APPENDIX E

LAUNCH BOOSTER EQUATIONS

An outline of the derivation of the launch booster equations and wind filter equations used in Section 6.3 is given in this Appendix. The derivation follows that in [6.1]. The vehicle equations model one axis of the booster, and have been linearized about a nominal trajectory. It is assumed that the vehicle is a rigid body, and that fuel-slosh and engine-inertia effects can be ignored.

The configuration of the vehicle and the relevant coordinates are shown in Figure E.1. Drift is measured along the y-axis from the nominal trajectory, and the pitch angle ϕ is measured from a reference direction along the trajectory.

The linearized drift equation is as follows (with β, ϕ, and α assumed small):

$$M \frac{d^2y}{dt^2} = (F_e - D_v)\phi + F_g\beta + \int_0^L \frac{dF_s}{d\alpha}\,\alpha\,d\ell, \qquad \text{(E-1)}$$

where M = vehicle mass (kg-sec^2/m),

y = drift from nominal trajectory (m),

F_e = total thrust of engines (kg),

D_v = vehicle drag (kg),

F_g = gimballed thrust (kg),

$\frac{dF_s}{d\alpha}$ = side force on missile per unit length per unit angle-of-attack (kg/m),

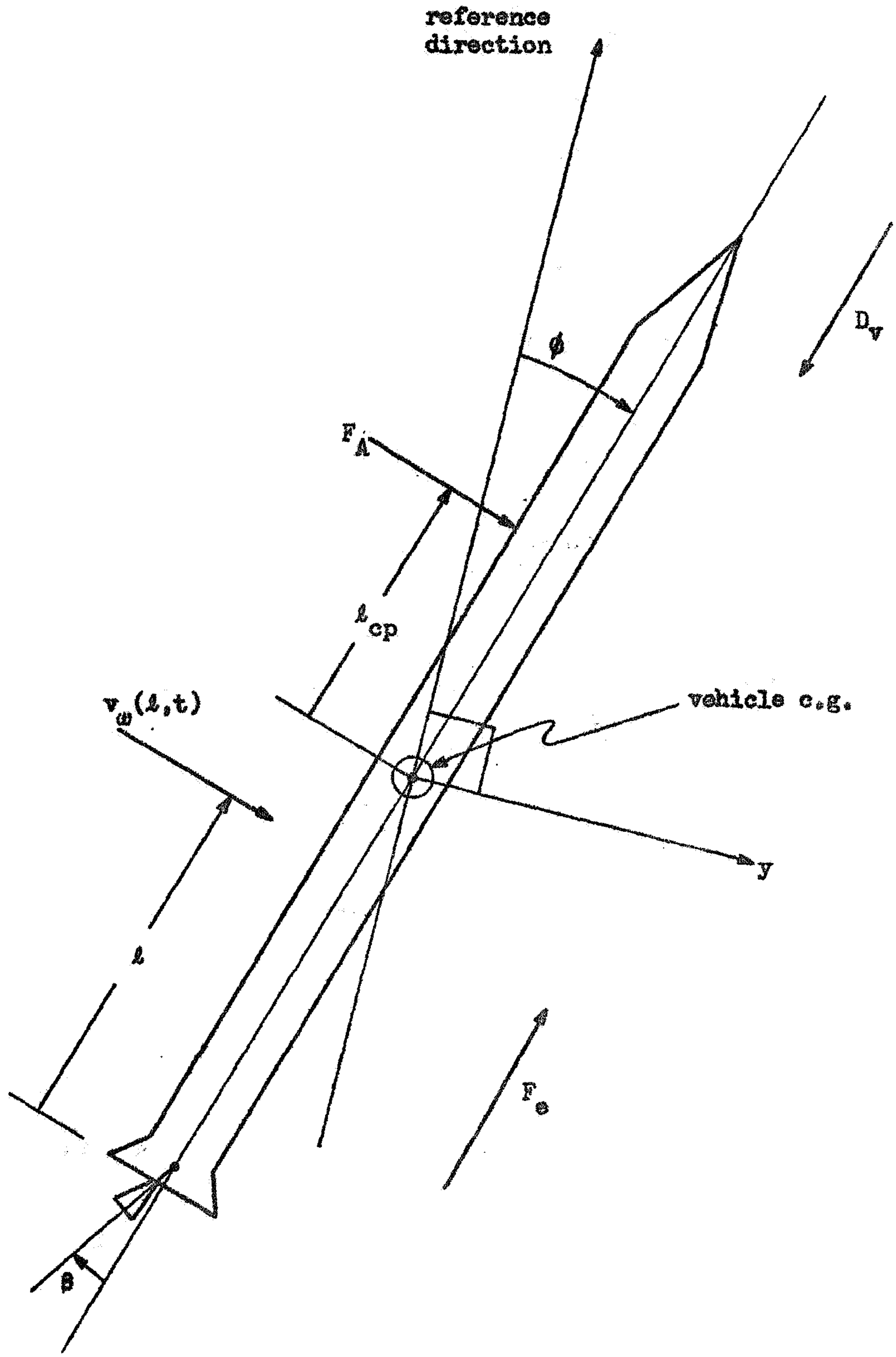


Figure E.1. Booster Model Configuration

$\alpha(\ell,t)$ = angle of attack at a distance ℓ from the tail of vehicle (rad),

L = vehicle length (m),

ℓ = distance from tail of vehicle (m),

ϕ = pitch angle deviation from reference direction along trajectory (rad),

β = engine gimbal angle (rad).

The angle-of-attack is given by:

$$\alpha(\ell,t) = \phi(t) + \frac{v_\omega(\ell,t) - \dot{y}(t) - [\ell-\ell_{cg}(t)]\dot{\phi}(t)}{V(t)}, \qquad \text{(E-2)}$$

where
$v_\omega(\ell,t)$ = wind velocity component orthogonal to vehicle at a distance ℓ from the vehicle tail (m/sec),

ℓ_{cg} = distance from tail to vehicle center of gravity (m),

$V(t)$ = nominal vehicle velocity (m/sec).

The pitch angle equation is given by:

$$I_p \frac{d^2\phi}{dt^2} = - F_g \ell_{cg} \beta + \int_0^L \frac{dF_s}{d\alpha} (\ell-\ell_{cg})\, \alpha d\ell, \qquad \text{(E-3)}$$

where I_p = pitch moment of inertia of vehicle (kg-m-sec^2).

Define the following terms:

$$F_A = \int_0^L \frac{dF_s}{d\alpha} d\ell \qquad \text{(E-4)}$$

= aerodynamic side force due to a unit angle-of-attack (kg),

$$F_A \ell_{cp} = \int_0^L \frac{dF_s}{d\alpha} (\ell-\ell_{cg}) d\ell \qquad \text{(E-5)}$$

= aerodynamic pitching moment due to a unit angle-of-attack (kg-m),

where ℓ_{cp} = aerodynamic moment arm (m).

$$T_{\dot{\phi}} = \int_0^L \frac{dF_s}{d\alpha} (\ell - \ell_{cg})^2 d\ell \tag{E-6}$$

= aerodynamic pitching moment due to a unit pitch rate for unit vehicle velocity (kg-m^2).

Then, substituting (E-2) into (E-1) and (E-3), and using the above definitions, we have (with the dots indicating time derivatives):

$$\ddot{y} = \frac{(F_e - D_v + F_A)}{M} \phi + \frac{F_g}{M} \beta - \frac{F_A}{MV(t)} \dot{y}(t)$$

$$- \frac{F_A \ell_{cp}}{MV(t)} \dot{\phi}(t) + \frac{1}{M} \int_0^L \frac{v_\omega(\ell,t)}{V(t)} \frac{dF_s}{d\alpha} d\ell \tag{E-7}$$

$$\ddot{\phi} = \frac{F_A \ell_{cp}}{I_p} \phi - \frac{F_g \ell_{cg}}{I_p} \beta - \frac{F_A \ell_{cp}}{I_p V(t)} \dot{y}(t)$$

$$- \frac{T_{\dot{\phi}}}{I_p V(t)} \dot{\phi} + \frac{1}{I_p} \int_0^L \frac{v_\omega(\ell,t)}{V(t)} \frac{dF_s}{d\alpha} (\ell - \ell_{cg}) d\ell \tag{E-8}$$

The structural bending moment (in kg-m) at a distance ℓ_o from the tail is given by:

$$I_b(\ell_o) = M_\beta \beta + \int_0^L \frac{dF_s}{d\alpha} [(\ell - \ell_{cg}) G_1 + G_2$$

$$+ (\ell_o - \ell) \mu (\ell_o - \ell)] \alpha d\ell, \tag{E-9}$$

where

$$M_\beta = F_g[\ell_o + G_2 - G_1 \ell_{cg}] \quad \text{(kg-m)},$$

$$G_1 = \frac{I_o + M_o(\ell_{ocg} - \ell_{cg})(\ell_{ocg} - \ell_o)}{I_p},$$

$$G_2 = \frac{M_o(\ell_{ocg} - \ell_o)}{M},$$

$$\mu(\ell_o-\ell) = \begin{cases} 1 \text{ if } 0 \leq \ell \leq \ell_o \\ 0 \text{ if } \ell > \ell_o \end{cases},$$

M_o = mass of section of vehicle from the tail to ℓ_o (kg),

ℓ_{ocg} = center of gravity of section of vehicle from tail to ℓ_o(m),

I_o = pitch moment of inertia of vehicle section from tail to ℓ_o about ℓ_{ocg} (kg-m-sec^2).

Define the following terms:

$$M_\alpha = \int_0^L \frac{dF_s}{d\alpha}[(\ell-\ell_{cg})G_1 + G_2 + (\ell_o-\ell)\mu(\ell_o-\ell)]d\ell \qquad \text{(E-10)}$$

= structural bending moment at ℓ_o due to a unit angle-of-attack (kg-m)

$$M_{\dot\phi} = \int_0^L \frac{dF_s}{d\alpha}[(\ell-\ell_{cg})G_1 + G_2 + (\ell_o-\ell)\mu(\ell_o-\ell)](\ell-\ell_{cg})d\ell \qquad \text{(E-11)}$$

= structural bending moment at ℓ_o due to a unit pitch rate (kg-m^2)

Then, substituting (E-2) into (E-9), and using the above definitions yields:

$$I_b(\ell_o) = M_\alpha\phi + M_\beta\beta - \frac{M_\alpha}{V(t)}\dot{y} - \frac{M_{\dot\phi}}{V(t)}\dot\phi + \int_0^L \frac{v_\omega(\ell,t)}{V(t)}\frac{dF_s}{d\alpha}[(\ell-\ell_{cg})G_1 + G_2 + (\ell_o-\ell)\mu(\ell_o-\ell)]d\ell. \qquad \text{(E-12)}$$

The integrals in equations (E-7), (E-8), and (E-12) must be evaluated. This was done in [6.1] by assuming that the incident wind loading could be represented by the output of a filter driven by $v_\omega(L,t)$. Define the load filter state variables to be η_1, η_2, and η_3(dimensionless). Th… the load filter equations are:

$$\dot{\eta}_1 = -\frac{V(t)}{H_1}\eta_1 + \frac{v_\omega(L,t)}{H_1} \tag{E-13}$$

$$\dot{\eta}_2 = -\frac{4V(t)}{H_2}\eta_2 - \frac{6V(t)}{H_2}\eta_3 - \frac{5v_\omega(L,t)}{H_2} \tag{E-14}$$

$$\dot{\eta}_3 = \frac{V(t)}{H_2}\eta_2 - \frac{v_\omega(L,t)}{H_2}\,. \tag{E-15}$$

where H_1 and H_2 are given constants (units of meters). Thus the integrals mentioned are approximated by:

$$\int_0^L \frac{v_\omega(\ell,t)}{V(t)}\,\frac{dF_s}{d\alpha}\,d\ell = F_A[a_1\eta_1 + a_2\eta_2] \tag{E-16}$$

$$\int_0^L \frac{v_\omega(\ell,t)}{V(t)}\,\frac{dF_s}{d\alpha}\,(\ell-\ell_{cg})d\ell = F_A[a_3\eta_1 - a_4\eta_2] \tag{E-17}$$

$$\int_0^L \frac{v_\omega(\ell,t)}{V(t)}\,\frac{dF_s}{d\alpha}\,[(\ell-\ell_{cg})G_1 + G_2 + (\ell_o - \ell)\mu(\ell_o - \ell)]d\ell = M_\alpha[a_5\eta_1 + a_6\eta_2]\,. \tag{E-18}$$

where the a_i's are given coefficients (a_1, a_2, a_5, and a_6 are dimensionless; a_3 and a_4 have dimensions of meters).

In [6.1], it was found that the incident wind, $v_\omega(L,t)$, could be represented by the output of another set of filter equations whose states are ω_1 and ω_2:

$$v_\omega(L,t) = \sigma_v \omega_1, \tag{E-19}$$

where $$\dot{\omega}_1 = V_h c_3 \omega_2 + c_1\sqrt{V_h}\, n(t) \tag{E-20}$$

$$\dot{\omega}_2 = - V_h c_5 \omega_1 - V_h c_4 \omega_2 + c_2\sqrt{V_h}\, n(t), \tag{E-21}$$

and $n(t)$ = white noise input with unit variance and zero mean

V_h = vertical component of vehicle velocity

c_1, c_2, c_3, c_4, c_5 = given coefficients.

In this booster model, the control u is a scalar which drives the gimballed engines. The equation describing the gimbal actuator dynamics is assumed to be:

$$\dot{\beta} = - 14.6\beta + 14.6u. \tag{E-22}$$

The bending-moment rate will be of interest when the response vector is formed.

Differentiate (E-12):

$$\begin{aligned}\dot{I}_b(\ell_o) = {} & \dot{M}_\alpha \phi + M_\alpha \dot{\phi} + \dot{M}_\beta \beta + M_\beta \dot{\beta} \\ & - \frac{d}{dt}\left(\frac{M_\alpha}{V(t)}\right)\dot{y} - \frac{M_\alpha}{V(t)}\ddot{y} - \frac{d}{dt}\left(\frac{M_\phi^*}{V}\right)\dot{\phi} \\ & - \frac{M_\phi^*}{V}\ddot{\phi} + \dot{M}_\alpha[a_5\eta_1 + a_6\eta_2] + M_\alpha a_5\dot{\eta}_1 \\ & + M_\alpha a_6\dot{\eta}_2 + M_\alpha \dot{a}_5\eta_1 + M_\alpha \dot{a}_6\eta_2\end{aligned} \tag{E-23}$$

Then, assuming that $\dot{V}(t)$, $\dot{a}_5$, and $\dot{a}_6$ are negligible, and substituting (E-7), (E-8), (E-13), (E-14), and (E-16) to (E-22) into (E-23) results in:

$$\dot{I}_b(\ell_o) = R_{\dot{y}}\dot{y} + R_{\phi}\phi + R_{\dot{\phi}}\dot{\phi} + R_{\beta}\beta + R_{\omega_1}\omega_1 + R_{\eta_1}\eta_1 + R_{\eta_2}\eta_2 + R_{\eta_3}\eta_3 + 14.6\, M_{\beta}u, \tag{E-24}$$

where

$$R_{\dot{y}} = -\frac{\dot{M}_{\alpha}}{V} + \frac{M_{\alpha}F_A}{MV^2} + \frac{M_{\dot{\phi}}F_A\ell_{cp}}{I_pV^2} \tag{E-25}$$

$$R_{\phi} = \dot{M}_{\alpha} - \frac{M_{\alpha}(F_e - D_v + F_A)}{MV} - \frac{M_{\dot{\phi}}F_A\ell_{cp}}{I_pV} \tag{E-26}$$

$$R_{\dot{\phi}} = M_{\alpha} + \frac{M_{\alpha}F_A\ell_{cp}}{MV^2} - \frac{\dot{M}_{\dot{\phi}}}{V} + \frac{M_{\dot{\phi}}T_{\dot{\phi}}}{I_pV^2} \tag{E-27}$$

$$R_{\beta} = \dot{M}_{\beta} - 14.6\, M_{\beta} - \frac{M_{\alpha}F_g}{MV} + \frac{M_{\dot{\phi}}F_g\ell_{cg}}{I_pV} \tag{E-28}$$

$$R_{\omega_1} = \frac{M_{\alpha}a_5\sigma_v}{H_1} - \frac{5M_{\alpha}\sigma_v a_6}{H_2} \tag{E-29}$$

$$R_{\eta_1} = \dot{M}_{\alpha}a_5 - \frac{M_{\alpha}F_A a_1}{MV} - \frac{M_{\dot{\phi}}F_A a_3}{I_pV} - \frac{M_{\alpha}Va_5}{H_1} \tag{E-30}$$

$$R_{\eta_2} = \dot{M}_{\alpha}a_6 - \frac{M_{\alpha}F_A a_2}{MV} + \frac{M_{\dot{\phi}}F_A a_4}{I_pV} - \frac{4M_{\alpha}Va_6}{H_2} \tag{E-31}$$

$$R_{\eta_3} = -\frac{6M_{\alpha}Va_6}{H_2} \tag{E-32}$$

It is convenient to summarize the above discussion by rewriting the vehicle equations, wind loading equations, and wind filter equations in the form of a set of first order linear differential equations. These equations can then be easily transformed to a state equation in vector-matrix form by letting the system states be y, $\dot{y}$, ϕ, $\dot{\phi}$, β, ω_1, ω_2, η_1, η_2, and η_3. The vehicle drift and pitch angle equations were found by substituting (E-16), (E-17), and (E-18) in (E-7) and (E-8).

<u>Launch Booster State Equations</u>

$$\frac{dy}{dt} = \dot{y} \tag{E-33}$$

$$\begin{aligned}\frac{d(\dot{y})}{dt} = &- \frac{F_A}{MV(t)}\dot{y}(t) + \frac{(F_e - D_v + F_A)}{M}\phi - \frac{F_A \ell_{cp}}{MV(t)}\dot{\phi} \\ &+ \frac{F_g}{M}\beta + \frac{F_A a_1}{M}\eta_1 + \frac{F_A a_2}{M}\eta_2\end{aligned} \tag{E-34}$$

$$\frac{d\phi}{dt} = \dot{\phi} \tag{E-35}$$

$$\begin{aligned}\frac{d(\dot{\phi})}{dt} = &- \frac{F_A \ell_{cp}}{I_p V(t)}\dot{y} + \frac{F_A \ell_{cp}}{I_p}\phi - \frac{T_{\dot{\phi}}}{I_p V(t)}\dot{\phi} \\ &- \frac{F_g \ell_{cg}}{I_p}\beta + \frac{F_A a_3}{I_p}\eta_1 - \frac{F_A a_4}{I_p}\eta_2\end{aligned} \tag{E-36}$$

$$\frac{d\beta}{dt} = -14.6\beta + 14.6u \tag{E-37}$$

$$\frac{d\omega_1}{dt} = V_h\omega_2 + c_1\sqrt{V_h}\, n(t) \tag{E-38}$$

$$\frac{d\omega_2}{dt} = - V_h c_5 \omega_1 - V_h c_4 \omega_2 + c_2\sqrt{V_h}\, n(t) \tag{E-39}$$

$$\frac{d\eta_1}{dt} = -\frac{V(t)}{H_1}\eta_1 + \frac{\sigma_v}{H_1}\omega_1 \tag{E-40}$$

$$\frac{d\eta_2}{dt} = -\frac{4V(t)}{H_2}\eta_2 - \frac{6V(t)}{H_2}\eta_3 - \frac{5\sigma_v}{H_2}\omega_1 \tag{E-41}$$

$$\frac{d\eta_3}{dt} = \frac{V(t)}{H_2}\eta_2 - \frac{\sigma_v}{H_2}\omega_1 \ . \tag{E-42}$$

The system responses considered in the example in Section 6.3 are y, $\dot{y}$, ξ, β, I_b, $\dot{\beta}$, and $\dot{I}_b$. Using the above derivations, the equations for these responses are:

Launch Booster Response Equations

$$r_1 = y \tag{E-43}$$

$$r_2 = \dot{y} \tag{E-44}$$

$$r_3 = \xi = -\frac{1}{V}\dot{y} + \phi + \frac{\sigma_v}{V}\omega_1 \tag{E-45}$$

$$r_4 = \beta \tag{E-46}$$

$$\begin{aligned} r_5 = I_b = &-\frac{M_\alpha}{V(t)}\dot{y} + M_\alpha\phi - \frac{M_{\dot{\phi}}}{V(t)}\dot{\phi} + M_\beta\beta \\ &+ M_\alpha a_5\eta_1 + M_\alpha a_6\eta_2 \end{aligned} \tag{E-47}$$

$$r_6 = \dot{\beta} = -14.6\beta + 14.6u$$

$$\begin{aligned} r_7 = \dot{I}_b = &R_{\dot{y}}\dot{y} + R_\phi\phi + R_{\dot{\phi}}\dot{\phi} + R_\beta\beta + R_{\omega_1}\omega_1 \\ &+ R_{\eta_1}\eta_1 + R_{\eta_2}\eta_2 + R_{\eta_3}\eta_3 + 14.6M_\beta u, \end{aligned} \tag{E-48}$$

where the R-coefficients in (E-48) are given in (E-25) to (E-32).

The numerical values of the coefficients used in the example in Section 6.3 were obtained from reference [6.1] and a NASA document, "Model Vehicle #2 For Advanced Control Studies." The latter document is a data package supplied by Marshall Space Flight Center, containing information on one model of a large flexible booster. The values of the constants used in the system and response equations are:

$$\left.\begin{aligned} &H_1 = 26.67,\ H_2 = 100,\ a_1 = a_2 = 1/2, \\ &c_1 = 1.378 \times 10^{-2},\ c_2 = -9.633 \times 10^{-7},\ c_3 = 1.0, \\ &c_4 = 1.9 \times 10^{-4},\ c_5 = 1.443 \times 10^{-8}. \end{aligned}\right\} \quad \text{(E-49)}$$

The values of the scalars a_5 and a_6 are given by:

$$\left.\begin{aligned} a_5(t) &= 1/3 + 2/3 \left(\frac{t}{150}\right) \\ a_6(t) &= 2/3 - 1/3 \left(\frac{t}{150}\right) \end{aligned}\right\} \quad \text{(E-50)}$$

The scalars a_3 and a_4 are defined by:

$$\left.\begin{aligned} a_3(t) - a_4(t) &= \ell_{cp}(t) \\ a_3(t) - 0.3a_4(t) &= \frac{P(t)}{F_A(t)}\ , \end{aligned}\right\} \quad \text{(E-51)}$$

where ℓ_{cp} and F_A are as defined above and $P(t)$ is a given time function.

The values of the other coefficients in the system and response equations are defined by Table E.1. This table gives the numerical values as a function of time of the quantities CF_1, CF_2,... CF_{22}, which are defined as:

Table E1

Launch Booster Coefficients

CF \ t (sec)	0	15	30	45	60	75
1	0.00E 00	-1.67E-03	-8.33E-03	-2.00E-02	0.00E 00	-6.00E-02
2	0.00E-00	-3.44E-05	-7.19E-05	-1.05E-04	0.00E-00	-1.25E-04
3	3.50E-01	3.67E-01	3.83E-01	4.00E-01	4.16E-01	4.67E-01
4	0.00E-00	2.33E-03	5.67E-03	9.83E-03	1.45E-02	1.17E-02
5	5.94E-00	6.55E-00	7.50E-00	8.13E-00	9.06E-00	1.03E 01
6	1.21E 01	1.33E 01	1.47E 01	1.63E 01	1.83E 01	2.05E 01
7	4.28E 07	4.22E 07	4.31E 07	4.37E 07	4.34E 07	4.31E 07
8	0.00E 00	5.00E 05	2.67E 06	7.83E 06	1.33E 07	1.73E 07
9	0.00E 00	1.16E 04	2.53E 04	4.21E 04	4.92E 04	3.42E 04
10	0.00E 00	7.20E 04	2.16E 05	4.32E 05	5.81E 05	-4.48E 05
11	0.00E 00	0.00E 00	1.33E 05	-4.41E 04	-2.21E 04	0.00E 00
12	0.00E 00	6.67E 04	1.63E 05	3.21E 05	5.08E 05	6.33E 05
13	2.00E-01	9.33E-02	4.92E-02	3.25E-02	3.20E-02	3.08E-02
14	0.00E-00	3.75E 01	1.02E 02	1.74E 02	2.63E 02	3.56E 02
15	4.24E 05	3.94E 05	3.65E 05	3.35E 05	3.05E 05	2.76E 05
16	-2.19E-07	-3.28E-07	-5.93E-07	-1.16E-06	-2.06E-06	-1.10E-07
17	0.00E 00	-3.91E-05	-6.87E-05	-1.25E-04	-7.50E-05	1.36E-04
18	0.00E 00	1.25E-02	2.66E-02	4.22E-02	5.56E-02	4.06E-02
19	0.00E 00	4.61E 01	1.07E 02	1.90E 02	2.83E 02	4.46E 02
20	0.00E 00	8.33E-02	4.01E-01	1.21E 00	2.77E 00	3.85E 00
21	0.00E 00	3.80E 06	8.90E 06	1.68E 07	2.75E 07	4.65E 07
22	2.85E 08	2.79E 08	2.75E 08	2.69E 08	2.63E 08	2.53E 08

CF \ t (sec)	90	105	120	135	150
1	-8.00E-02	-4.02E-02	-1.83E-02	-5.00E-03	0.00E-00
2	-1.09E-04	-4.06E-05	-1.25E-05	0.00E-00	0.00E-00
3	5.17E-01	5.83E-01	7.00E-01	9.17E-01	1.58E-00
4	7.00E-03	2.83E-03	1.00E-03	3.33E-04	0.00E-00
5	1.21E 01	1.38E 01	1.63E 01	1.94E 01	2.38E 01
6	2.33E 01	2.70E 01	3.17E 01	3.77E 01	4.63E 01
7	4.59E 07	4.81E 07	4.61E 07	4.72E 07	7.28E 07
8	8.67E 06	5.17E 06	1.67E 06	8.33E 05	3.33E 05
9	1.21E 04	6.33E 03	1.05E 03	5.25E 02	0.00E 00
10	-1.28E 05	-3.44E 05	-9.59E 04	-4.80E 04	0.00E 00
11	3.33E 05	0.00E 00	-8.91E 04	4.44E 05	2.59E 06
12	1.17E 05	3.36E 04	8.36E 03	5.00E 02	0.00E 00
13	1.33E-02	6.67E-03	6.67E-03	2.50E-03	0.00E-00
14	4.56E 02	5.44E 02	6.42E 02	7.34E 02	8.56E 02
15	2.46E 05	2.16E 05	1.87E 05	1.57E 05	1.28E 05
16	7.80E-08	1.41E-07	3.10E-08	0.00E 00	0.00E 00
17	2.34E-05	-6.25E-06	0.00E 00	0.00E 00	0.00E 00
18	2.16E-02	8.17E-03	2.50E-03	6.25E-04	0.00E 00
19	6.46E 02	9.27E 02	1.28E 03	1.72E 03	2.34E 03
20	4.18E 00	2.68E 00	1.43E 00	5.00E-01	1.85E-01
21	3.95E 07	2.94E 07	2.32E 07	1.85E 07	1.45E 07
22	2.44E 08	2.28E 08	2.06E 08	1.74E 08	1.22E 08

$$CF_1 = \frac{F_A \ell_{cp}}{I_p} \qquad CF_2 = \frac{F_A \ell_{cp}}{I_p V} \qquad CF_3 = \frac{F_g \ell_{cg}}{I_p}$$

$$CF_4 = \frac{F_A}{MV} \qquad CF_5 = \frac{F_g}{M} \qquad CF_6 = \frac{F_e - D_v}{M}$$

$$CF_7 = M_\beta \qquad CF_8 = M_\alpha \qquad CF_9 = \frac{M_\alpha}{V}$$

$$CF_{10} = \dot{M}_\alpha \qquad CF_{11} = \dot{M}_\beta \qquad CF_{12} = \frac{M_\alpha \sigma_v}{V} \tag{E-52}$$

$$CF_{13} = \frac{\sigma_v}{V} \qquad CF_{14} = V_h \qquad CF_{15} = M$$

$$CF_{16} = \frac{M_{\dot{\phi}}}{V} \qquad CF_{17} = \frac{\dot{M}_{\dot{\phi}}}{V} \qquad CF_{18} = \frac{T_{\dot{\phi}}}{I_p V}$$

$$CF_{19} = V \qquad CF_{20} = \frac{F_A}{M} \qquad CF_{21} = P$$

$$CF_{22} = I_p \; .$$

The values of the CF_i for intermediate points in time were determined by linear interpolation, using the given values in Table E.1.

APPENDIX F

A "BOUNDED-RESPONSE" STOCHASTIC CONTROL PROBLEM

This Appendix outlines what can be called the "bounded-response" stochastic control problem, which was discussed by Skelton in [6.1]. The performance index derived here was used in the second example in Chapter 6 to evaluate the performance of the launch booster controls.

An ℓ-dimensional response vector r was defined in (2-3). In this problem, it is desirable that the magnitude of the ith component of r be bounded by a given value, say γ_i. Since r(t) is a Gaussian random variable with a nonzero variance, it is not meaningful to place a hard constraint on r. So the constraint on r must be a probabilistic one. To formulate the bounded-response stochastic problem, first define the following events:

$$a_i = \{\text{event that } |r_i(T)| < \gamma_i\} \text{ , } i = 1,2,\ldots k \tag{F-1}$$

$$\begin{aligned} b_i(j) = \{&\text{event that } |r_i(t)| = \gamma_i \ \underline{\text{and}} \\ &\frac{d}{dt}|r_i(t)| > 0 \ \underline{\text{exactly}}\ j \text{ times in} \\ &[t_o,T)\} \text{ , } i = k+1,\ldots \ell \text{ ,} \end{aligned} \tag{F-2}$$

where the γ_i are positive real numbers, and T is the given terminal time for the problem. Note that the first k responses are to be bounded at the terminal time only, and the other responses are to be bounded during the time interval of interest. A probabilistic performance index for this problem is then defined as

$$\bar{J} = 1 - \Pr\{a_1, a_2, \ldots, a_k, b_{k+1}(0), \ldots b_\ell(0)\} \tag{F-3}$$

= probability that at least one terminal response falls outside its bound or at least one of the last (ℓ-k) responses falls outside its bound at least once for $t \in [t_o, T)$.

Then the bounded-response problem is that of finding the $u \in U$ (defined in (2-9)) such that $\bar{J}$ is minimized, subject to the system side-conditions (2-1) to (2-8) and the Kalman filter side-conditions (2-10) to (2-15).

The problem as stated above is a difficult one, and has not been solved to date. However, it is possible to find an upper bound to the performance index in (F-3), and the problem of minimizing this upper bound is a simpler one. Let

$$N_i = \left\{ \begin{matrix} \text{number of times } |r_i(t)| > \gamma_i, \\ t \in [t_o, T),\ i = k+1, \ldots \ell \end{matrix} \right\}. \tag{F-4}$$

Then we can define a new performance index

$$J_s = \sum_{i=1}^{k} \Pr(\bar{a}_i) + \sum_{i=k+1}^{\ell} E[N_i] , \tag{F-5}$$

where a_i is defined in (F-1), $\bar{a}_i$ is the event that a_i does not occur, and Pr ( ) is the probability operator. It can then be shown (see [6.1]) that for a general stochastic process, J_s is an upper bound for $\bar{J}$:

$$J_s \geq \bar{J} . \tag{F-6}$$

The advantage of minimizing J_s instead of $\bar{J}$ is that an explicit expression for J_s can be written in terms of the response covariance matrix S (defined in (2-17)). Also, it can be seen from (F-5) that J_s is itself a meaningful performance index; so we can have some assurance that minimizing J_s will result in reasonable system performance. Of course, the optimal J_s must be small for it to be a meaningful upper bound to the probability of the event in (F-3).

To express J_s in the simplest form, we require that the response ~~be~~ formed in the following manner:

$$r = [r_1 \dots r_k r_{k+1} \dots r_{\ell_1} r_{\ell_1+1} \dots r_\ell]' \ . \qquad \text{(F-7)}$$

where $(\ell_1 - k) = (\ell - \ell_1)$, and

$r_1, r_2, \dots r_k$ = responses which are to be controlled at the terminal time,

$r_{k+1}, r_{k+2}, \dots r_{\ell_1}$ = responses which are to be controlled for time $t \in [t_o, T)$,

$r_{\ell_1+1}, r_{\ell_1+2}, \dots r_\ell = \dot{r}_{k+1}, \dot{r}_{k+2}, \dots \dot{r}_\ell$

= uncontrolled responses which give values of $\dot{r}_i(t)$, $i = k+1, \dots \ell_1$.

The first ℓ_1 responses are the only ones of actual interest. The last $(\ell_1 - k)$ responses are uncontrolled, which is equivalent to setting γ_i arbitrarily large for $i = \ell_1 + 1, \dots \ell$. Thus

$$E[N_i] = 0, \ i = \ell_1 + 1, \dots \ell \ . \qquad \text{(F-8)}$$

So (F-5) can be rewritten

$$J_s = \sum_{i=1}^{k} \Pr(\bar{a}_i) + \sum_{i=k+1}^{\ell_1} E[N_i] , \qquad \text{(F-9)}$$

which indicates that the last (ℓ_1-k) responses are not directly considered in the performance index. The reason for including them in the response vector is that both $r_i(t)$ and $\dot{r}_i(t)$, $i = k+1,\ldots\ell_1$, must be checked to see if the event $b_i(j)$ occurs as defined in (F-2). Specifically, $E[N_i]$ for $i = k+1,\ldots\ell_1$ is a function of the quantities $E[r_i^2(t)]$, $E[r_i(t)\dot{r}_i(t)]$, and $E[\dot{r}_i^2(t)]$, where $E[\cdot]$ denotes expectation.

The complete expression for J_s can be derived (as in [6.1]) by extending the Rice zero-crossing formula for stationary Gaussian processes given in [F.1]. The resulting expression is:

$$J_S = g_1[S(T)] + \int_{t_o}^{T} g_2[S(t)]dt, \qquad \text{(F-10)}$$

where $S(t) = E[r(t)r'(t)]$, (F-11)

$$g_1[S(T)] = \sum_{i=1}^{k} 2\,\Phi_N\left(-\frac{\gamma_i}{\sqrt{S_{ii}(T)}}\right) \qquad \text{(F-12)}$$

$$g_2[S(t)] = \sum_{i=k+1}^{\ell_1} 2\,P_i(t) , \qquad \text{(F-13)}$$

$$\Phi_N(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-y^2/2}\,dy, \qquad \text{(F-14)}$$

and

$$P_i(t) = \frac{\exp\left[-\gamma_i^{\,2}/2M_{1i}\right]}{\sqrt{2\pi M_{1i}}}\left\{\frac{\sigma_i \exp\left[-\rho_i^{\,2}/2\,\sigma_i^2\right]}{\sqrt{2\pi}} - \rho_i\left[1-\Phi_N\left(\frac{\rho_i}{\sigma_i}\right)\right]\right\} , \qquad \text{(F-15)}$$

$$\rho_i = -\frac{\gamma_i M_{ci}}{M_{1i}} , \quad \sigma_i = \left[M_{2i} - \frac{M_{ci}^2}{M_{1i}}\right]^{1/2} , \qquad \text{(F-16)}$$

$$\left.\begin{aligned} M_{1i} &= E[r_i^2(t)] , \\ M_{ci} &= E[r_i(t)\dot{r}_i(t)] = E[r_i(t)r_{i+\ell_1-k}(t)] , \\ M_{2i} &= E[\dot{r}_i^2(t)] = E[r_{i+\ell_1-k}^2(t)] . \end{aligned}\right\} \quad \text{(F-17)}$$

So J_s is a function of the covariance matrix S, and is a special case of the performance index J in (2-16). Therefore, the problem of minimizing J_s is a special case of the general problem stated in Section 2.2.

Expressions for the partial derivative matrices $\frac{\partial g_1}{\partial S}(T)$ and $\frac{\partial g_2}{\partial S}(t)$ will be given below. These expressions are needed to form the gradient of J_s (from the definition in (3-20)),

$$DJ_s(\hat{s}) = \left[\frac{\partial g_1}{\partial s}(T), \frac{\partial g_2}{\partial s}(t)\right]\Bigg|_{\hat{s}} ,$$

which is used in the computational algorithms in Section 6.3. The vectors $\frac{\partial g_1}{\partial s}(T)$ and $\frac{\partial g_2}{\partial s}(t)$ are formed from the corresponding matrices by the "stacking" procedure outlined in section 3.4. We have:

$$\frac{\partial g_1}{\partial S_{ij}} = \begin{cases} \dfrac{\gamma_i \exp\left(-\gamma_i^2/2S_{ii}\right)}{\sqrt{2\pi S_{ii}}\, S_{ii}} , & \text{if } i = j \text{ and } i \leq k \\ 0 , & \text{other } i,j. \end{cases} \quad \text{(F-18)}$$

The matrix $\frac{\partial g_2}{\partial S}$ is expressed in the following way. The only nonzero elements in the matrix are $\frac{\partial g_2}{\partial S_{mm}}$, $\frac{\partial g_2}{\partial S_{mn}}$, $\frac{\partial g_2}{\partial S_{nm}}$, and $\frac{\partial g_2}{\partial S_{nn}}$, where $m = k+1, k+2, \ldots \ell_1$, and $n = m+(\ell_1-k)$. The above elements are:

$$\frac{\partial g_2}{\partial S_{mm}} = P_m h_1 + h_2\left[\frac{S_{mn}^2}{2\sigma_m S_{mn}^2} h_3 - \frac{\gamma_m S_{mn}}{S_{mm}^2} h_4\right] \tag{F-19}$$

$$\frac{\partial g_2}{\partial S_{mn}} = -h_2\left[\frac{S_{mn}}{\sigma_m S_{mm}} h_3 - \frac{\gamma_m}{S_{mm}} h_4\right] \tag{F-20}$$

$$\frac{\partial g_2}{\partial S_{nm}} = \frac{\partial g_2}{\partial S_{mn}} \tag{F-21}$$

$$\frac{\partial g_2}{\partial S_{nn}} = \frac{h_2 h_3}{2\,\sigma_m}, \tag{F-22}$$

where P_m is defined in (F-15), and

$$h_1 = \frac{\gamma_m^2 - S_{mm}}{S_{mm}^2}, \tag{F-23}$$

$$h_2 = \frac{2\exp\left(-\gamma_m^2/2S_{mm}\right)}{\sqrt{2\pi}\,S_{mm}}, \tag{F-24}$$

$$h_3 = \frac{\exp\left(-\rho_m^2/2\sigma_m^2\right)}{\sqrt{2\pi}}, \tag{F-25}$$

$$h_4 = 1 - \Phi_N(\rho_m/\sigma_m). \tag{F-26}$$

The quantities Φ_N, ρ_m, and σ_m are defined in (F-14) and (F-16).

As an example of the above procedure, suppose r is formed such that $k = 3$, $\ell_1 = 5$, and $\ell = 7$. Then the partial derivative matrices have the form:

$$\frac{\partial g_1}{\partial s} = \begin{bmatrix} \frac{\partial g_1}{\partial s_{11}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial g_1}{\partial s_{22}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial g_1}{\partial s_{33}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\frac{\partial g_2}{\partial s} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial g_2}{\partial s_{44}} & 0 & \frac{\partial g_2}{\partial s_{46}} & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial g_2}{\partial s_{55}} & 0 & \frac{\partial g_2}{\partial s_{57}} \\ 0 & 0 & 0 & \frac{\partial g_2}{\partial s_{64}} & 0 & \frac{\partial g_2}{\partial s_{66}} & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial g_2}{\partial s_{75}} & 0 & \frac{\partial g_2}{\partial s_{77}} \end{bmatrix}$$

and the elements are computed as indicated above.

APPENDIX G

COMPUTATIONAL TECHNIQUES

The programming methods and computational techniques described in this Appendix were used in the two example problems discussed in Chapter 6. The philosophy used in writing the programs was to keep the computational work as simple as possible, subject to the accuracy requirements of the examples and the considerations of computational time. The good results obtained in the examples in Chapter 6 indicate that the techniques used were quite adequate for the purpose of illustrating the PGM and DGIM algorithms.

Programming was done using the FORTRAN IV language, and the programs were run on the IBM 7094 (for example 1) and CDC 6500 (for example 2) computing systems.

G.1 Solution of Differential Equations

The implementation of the PGM and DGIM algorithms included the task of solving several nonlinear matrix differential equations. These equations are as follows:

a) the matrix Riccati equation, which is given by equations (2-23), (2-24), and the definition of $K^*(t)$ in (2-22). Given the parameter matrices A, B, C, and D, and the quadratic coefficient matrices $Q(t)$ and $Q_F(T)$, the Riccati matrix $P_v(t)$ and the optimal feedback coefficient $K^*(t)$ had to be computed. In the description of the algorithms in Figures 5.1 and 5.3, this procedure was called "Solve the $\hat{q}_i$

problem, yielding $K_i^*(t)$." The function space element $\hat{q} = [q_F, q(t)]$ defines the matrices $Q_F(T)$ and $Q(t)$, as is noted in Section 3.4, and $K_i^*(t)$ is defined in (2-22). The subscript i refers to the ith stage of the iteration process.

b) the error covariance equation given in (2-14). This equation had to be solved for $E_k(t)$, which is a matrix function needed in the state covariance equation (3-3) and the response covariance equation (3-1).

c) the state-covariance equation given by equation (3-3). This equation was used in the procedure in the algorithms entitled, "Find $\hat{s}_i^*$ resulting from $K_i^*(t)$." First, $K_i^*(t)$ was used in (3-3) (in place of K(t)) to obtain $C_x(t)$. The matrix $E_k(t)$ in (3-3) was found in part b) above. Then, $C_x(t)$ and $K_i^*(t)$ were both used in equation (3-1) along with $E_k(t)$ to obtain the response covariance matrix $S_i^*(t)$, which was then converted to $\hat{s}_i^*$ by the "stacking" procedure described in section 3.4. The "*" in the above discussion refers to the solution of a J_Q-problem, and the subscript i refers to the ith stage of the iteration process.

In both examples, the above equations were solved by using digital numerical integration techniques. In the first example, a fourth-order Runge-Kutta method with a fixed step size of 0.02 seconds was used (the total time interval was 10 seconds). This method resulted in a computer time-per-iteration of 75 sec. for PGM and 40 sec. for DGIM on the IBM 7094. These iteration times include the numerical integration of the above equations and the computation of the new Q(t) and Q_F matrices. The iteration times were acceptable, as was the accuracy of the integration technique (as measured by halving the step size, rerunning a few iterations, and comparing the results). So no further refinement of technique was attempted.

The task of solving the differential equations in the second example was a much more difficult one, because the high order of the system and the time-varying nature of the coefficients resulted in extremely long integration times, even on the CDC 6500. The integration method finally decided upon after much experimentation was the simple Euler method (linear extrapolation of the derivative), with a fixed step size of 0.01 seconds. The total problem time was 150 seconds. Other integration methods, such as Runge-Kutta and Hamming predictor-corrector were tried, but they took two to four times as much computational time as the Euler method, using the same basic step size. (The computational time needed to integrate the various differential equations is discussed below.) It was found that the results of the numerical integrations using the three methods mentioned above were quite comparable when the basic step size of 0.01 seconds was used in each. Therefore, the Euler method was chosen because of its speed and simplicity. Skelton used a modified version of the Euler method on the same problem in [6.1], and also found that it was an adequate integration technique.

Several techniques for saving computer time were used in the integration routines:

a) The matrix equations to be integrated were the state covariance equation (3-3) and the Riccati equation (2-23). (In the second example, the Kalman filter equations were not required; so the error covariance matrix $E_k(t)$ did not have to be computed, and the terms in (3-1) and

(3-3) involving E_k were set to zero.) The solution matrices ($C_x(t)$ and $P_v(t)$) of these equations are both symmetric, so only the upper half and the diagonal parts of the matrices were computed. Since both P_v and C_x were 10 by 10 matrices, this meant that 55 simultaneous equations (instead of 100) were integrated in each case.

b) As can be seen from (2-23) and (3-3), the computation of the derivative matrices $\dot{P}_v(t)$ and $\dot{C}_x(t)$ involved a number of multiplications of high-order matrices. These multiplications were the operations in the integration routine which took the most computer time. Therefore, special routines which eliminated many zero-multiplications were written and used, instead of standard matrix multiplication subroutines.

c) It can be seen from the system equations as written in Appendix E (equations (E-33) to (E-42)), that the parameter matrices A,B,C, and D must be recomputed every 0.01 seconds during the integration process, using the given (or interpolated) values of the CF_i-coefficients. To save on computer time, the values of the A,B,C, and D matrices at five-second intervals were instead computed beforehand. Then their given or linearly interpolated values were used directly in the integration routines and in the computation of S(t).

Using the above techniques and the Euler integration method resulted in reasonable computer times (central processor execution time on the CDC 6500) in the second example. The integration of the Riccati equation in (2-23) took 310 seconds; the integration of the covariance equation in (3-3), together with the computation of S by (3-1) took 445 seconds. It should be mentioned that the performance indices J_s in (6-34) and J_N in (6-40) involved integrals of functions of S(t), and were therefore computed along with $C_x(t)$ and $S(t)$. The computation of J_N took about 10 seconds, and that of J_s took about 115 seconds (due to the complexity of J_s). It was this difference in time required to compute J_N and J_s that was the main reason for the difference in computer times for the PGM and DGIM algorithms, as mentioned in Section 6.3.

It is possible that a hybrid computer facility would have been the most efficient computing tool for the implementation of the PGM and DGIM algorithms. The great bulk of the digital computer time was used to integrate the Riccati and covariance equations. Much time could have been saved if the equations were integrated on an analog computer. The computation of the new quadratic coefficients would have been performed digitally. Since no hybrid facility was available, however, it was not possible to try this computational method.

G.2 Storage and Handling of Time Functions

When implementing the PGM and DGIM algorithms, it was often necessary to store, punch out, or manipulate certain matrix time functions defined on the entire problem time interval. For example, a feedback coefficient matrix $K(t)$, $t \in [t_o, T]$, which was computed by integrating the Riccati equation (2-23), had to be stored on punch cards so that it could be used later in computing $C_x(t)$ from equation (3-3). The method used was to store the values of the matrices at given equidistant time instants, and use these values in manipulations or to produce punch card output. Then the resulting matrix time function was recovered (approximately) by interpolating the given values. Linear interpolation was generally used, because higher-order interpolation methods could not be justified unless extensive knowledge of the time functions involved was available. This knowledge was not available beforehand, in general, so the linear interpolation method was used. In the first example, the values of the functions were stored at 1-second intervals (over a 10-second problem time) and in the second example, at 5-second intervals (over a 150-second problem time). The above intervals were chosen experimentally, and were found to be adequate.

G.3 One-dimensional Minimization in PGM

The crucial step in the PGM algorithm, as described in Figure 5.3, is the determination of the $\gamma \in [0,1]$ at which $J[(1-\gamma)\hat{s}_i + \gamma \hat{s}_i^*]$ is a minimum, given two points $\hat{s}_i$, $\hat{s}_i^* \in \sigma$. This can be viewed as a "one-dimensional" minimization problem, in which the functional J is to be minimized on the "straight line" connecting $\hat{s}_i$ and $\hat{s}_i^*$. The technique used in performing this task was based on the fact that the functionals to be minimized in both examples were convex. As described in Chapter 6, the PGM algorithm was applied to J in (6-6) in the first example, and to J_N in (6-40) in the second example. Because of this convexity property, a local minimum point along the "straight line" is also the absolute minimum point. Therefore, the minimization technique was simply to "walk" from $\hat{s}_i$ to $\hat{s}_i^*$, sampling the functional J along the way, until a local minimum of J was found.

To implement this technique, a method of taking appropriate "steps" along the line was developed, as was a method for evaluating J at each step. These two methods will be described separately:

1) "Walking" Technique

At the ith stage of the PGM algorithm, the points $\hat{s}_i$ and $\hat{s}_i^*$ are available. A method of "walking" on a straight line from $\hat{s}_i$ to $\hat{s}_i^*$

to find an approximate local minimum of J on this line is defined in Figure G.1. The following notation will be used:

L_i = "straight line" from $\hat{s}_i$ to $\hat{s}_i^*$

$$= \left\{ \begin{matrix} \hat{s}: \hat{s} = (1-\gamma)\hat{s}_i + \gamma\hat{s}_i^*, \\ \text{for all } \gamma \in [0,1] \end{matrix} \right\}$$

γ = fractional distance along L_i from $\hat{s}_i$ to $\hat{s}_i^*$, as in the above definition of L_i

The basic idea in the walking technique is to store the value of J at the beginning of the line, at $\hat{s}_i$. Then a "step" along L_i is taken, and the value of J is sampled at this new point. If the latter value of J is less than that at $\hat{s}_i$, then the new point is stored and the "walk" is continued as long as the sampled values of J continue to decrease. If the sampled value of J at a new point is greater than at the previous point, the previous point is taken to be an approximate local minimum point of J on L_i.

During the process of the walk, let

$\hat{s}_{im}$ = the point on L_i at which J has taken on its smallest value so far,

γ_m = the value of γ which corresponds to $\hat{s}_{im}$,

$\hat{s}_{ic}$ = the next point on L_i at which the value of J is to be sampled and compared with $J(\hat{s}_{im})$,

γ_c = the value of γ which corresponds to $\hat{s}_{ic}$,

$\Delta\gamma$ = the size of the "step" to be taken from $\hat{s}_{im}$ to $\hat{s}_{ic}$, measured as a fraction of the length of L_i.

Using the above notation, the procedure in Figure G.1 can be described by the following sequence of steps:

a) The walk begins at $\hat{s}_i$. So $\hat{s}_{im}$ will be set equal to $\hat{s}_i$, and

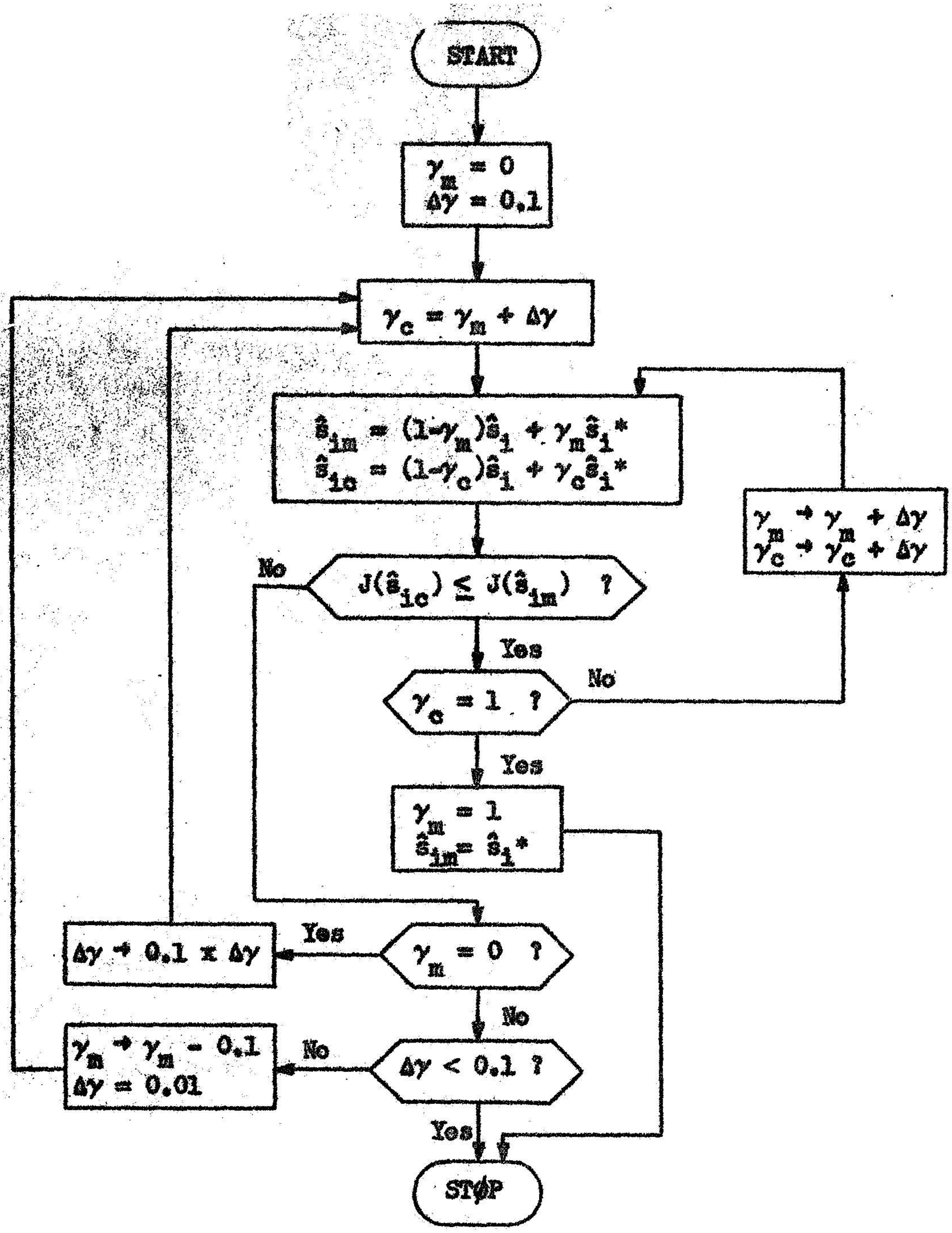


Figure G.1 One-Dimensional Minimization Technique

$\gamma_m = 0$ initially. The initial step size is $\Delta\gamma = 0.1$; that is, the length of the first step is one-tenth the distance from $\hat{s}_i$ to $\hat{s}_i$*.

b) The point $\hat{s}_{im}$ is computed using the given value of γ_m. The point $\hat{s}_{ic}$, at which J will be evaluated and compared to $J(\hat{s}_{im})$, is also computed, using $\gamma_c = \gamma_m + \Delta\gamma$. (At the beginning of the walk, $\gamma_m = 0$ and $\gamma_c = 0.1$; so $\hat{s}_{im} = \hat{s}_i$ and $\hat{s}_{ic} = 0.9\ \hat{s}_i + 0.1\ \hat{s}_i$*.)

The computation of $\hat{s}_{im}$ and $\hat{s}_{ic}$ is carried out as follows. The points $\hat{s}_i$ and $\hat{s}_i$* are defined by the corresponding matrix functions of time $S^i(t)$ and $S^{i*}(t)$ by the "stacking" procedure described in section 3.4 (where the superscript i corresponds to the subscript i in $\hat{s}_i$). In the actual computation process, $S^i(t)$ and $S^{i*}(t)$ are available in the form of values of the matrices at equidistant time instants in the interval $[t_o,T]$, say at $t = t_k$, $k = 0,1,2,\ldots\bar{k}$, where $t_{\bar{k}} = T$, and $t_{k+1}-t_k$ is constant. The intermediate values of the matrices are then approximated by linear interpolation (this method of storing and handling matrix time functions is described in section G.2). Let the points $\hat{s}_{im}$ and $\hat{s}_{ic}$ be defined by the corresponding matrix time functions $S^{im}(t)$ and $S^{ic}(t)$. The values of these matrices at $t=t_k$, $k=0,1,2,\ldots\bar{k}$ were computed by:

$$S^{im}(t_k) = (1-\gamma_m)S^i(t_k) + \gamma_m S^{i*}(t_k) \qquad \text{(G-1)}$$

$$S^{ic}(t_k) = (1-\gamma_c)S^i(t_k) + \gamma_c S^{i*}(t_k) \qquad \text{(G-2)}$$

for $k = 0,1,2,\ldots\bar{k}$.

The intermediate values of $S^{im}(t)$ and $S^{ic}(t)$ in the interval $[t_o,T]$ are then defined by linear interpolation of the values of the matrices at the discrete times. By the above procedure, the entire time functions $S^{im}(t)$ and $S^{ic}(t)$ are defined, and therefore so are $\hat{s}_{im}$ and $\hat{s}_{ic}$. The discrete time instants used in each example were discussed in section 6.2.

c) The values of J at $\hat{s}_{im}$ and $\hat{s}_{ic}$ are compared; if $J(\hat{s}_{ic}) \leq J(\hat{s}_{im})$, then γ_m is set equal to γ_c and the walk continues. The procedure returns to b), with updated γ_m and γ_c. If the above inequality holds for $\gamma_c = 1.0$, this means that the "walk" from $\hat{s}_i$ to $\hat{s}_i^*$ has been completed, and that $\hat{s}_i^*$ is the approximate minimum point of J on L_i. So γ_m is set equal to 1, $\hat{s}_{im}$ is set equal to $\hat{s}_i^*$, and the minimization procedure is terminated.

d) If the inequality $J(\hat{s}_{ic}) > J(\hat{s}_{im})$ holds instead of the reverse inequality in c), then $\hat{s}_{im}$ is an approximate local minimum point along L_i, within the accuracy of the step size $\Delta\gamma = 0.1$.

e) If the inequality in d) occurs on the first "step" (i.e., when $\gamma_m = 0$ and $\gamma_c = 0.1$), it means that the minimum point of J on L_i must occur for $\gamma \in (0, 0.1)$. Therefore, the "step size" $\Delta\gamma$ is reduced by a factor of 10, and the "walk" is restarted at $\hat{s}_i$ (with $\gamma_m = 0$). This reduction of step size and restarting of the walk is repeated until an approximate minimum point is found in $(0, 0.1)$. A minimum point is guaranteed to exist by conclusion 1) of Theorem 5.1, so an approximate minimum point can be found.

f) At this stage in the procedure, an approximate minimum point of J has been found by the steps outlined above. This point is one of three types: i) it is the end point of the line; i.e., $\gamma_m = 1$ and $\hat{s}_{im} = \hat{s}_i^*$; ii) it is a point in $(0, 0.1)$, and has been determined with an accuracy corresponding to the value of $\Delta\gamma$ when the approximate local minimum is found (remember that $\Delta\gamma$ is divided by 10 until such a minimum is found); iii) otherwise, it is a point $\hat{s}_{im}$ defined by γ_m=0.1 x N, where N is some integer from 1 through 9, and has been found within the accuracy of the step size $\Delta\gamma = 0.1$. In this last case, a better approximation is then computed by letting $\gamma_m \to \gamma_m - 0.1$ (i.e., by taking a "step backward"), reducing the step size to $\Delta\gamma = 0.01$, and restarting the "walk" from the point $\hat{s}_{im}$ which corresponds to the new γ_m. A new approximate minimum point is then determined with an accuracy corresponding to a step size of $\Delta\gamma = 0.01$.

The procedure described above was the one used in the first example in Chapter 6. In the second example, the reduction of step size to $\Delta\gamma = 0.01$ and the subsequent refinement of the approximate minimum point (described above in f)) was not performed. This simplification was made because it was found (in the first example) that the refinement procedure did not speed up the convergence process to any great extent. The results described in Section 6.3 using the simplified minimization procedure were quite satisfactory, so no further adjustments in the procedure were made.

2) Evaluation of J

One of the steps in the minimization procedure described above requires that $J(\hat{s}_{im})$ and $J(\hat{s}_{ic})$ be evaluated, given the points $\hat{s}_{im}$ and $\hat{s}_{ic}$. The points $\hat{s}_{im}$ and $\hat{s}_{ic}$ are defined by the corresponding matrix time functions, $S^{im}(t)$ and $S^{ic}(t)$, as described in step b) of the above minimization procedure. In the following discussion, it is assumed that the values of $S^{im}(t)$ and $S^{ic}(t)$ at discrete instants of time have been computed, using equations (G-1) and (G-2). The values of the matrices at intermediate points in time are found by linear interpolation, as mentioned previously.

The method used to evaluate $J(\hat{s}_{im})$, given $S^{im}(t)$ in the above form, will now be described ($J(\hat{s}_{ic})$ was found in a similar way). The general form of J used in the evaluation procedure is given in (2-16). Since $S^{im}(T)$ was given by (G-1), the first term in (2-1) was computed directly. The second term was evaluated by first defining a new variable:

$$p_{im}(t) = \int_{t_o}^{t} f_2[S^{im}(\tau)]d\tau \ , \qquad \text{(G-3)}$$

where f_2 is the same as in (2-16). A corresponding differential equation for $p_{im}(t)$ is:

$$\frac{dp_{im}(t)}{dt} = f_2[S^{im}(t)] \ , \qquad \text{(G-4)}$$

with $p_{im}(t_o) = 0$. (G-5)

The above differential equation was then integrated numerically, and the value of the second term in (2-21) was set equal to $p_{im}(T)$ to complete the computation of $J(\hat{s}_{im})$.

The numerical integration methods used in the evaluation of J were the same as used to integrate the Riccati and covariance matrix equations (see section G.1). A Runge-Kutta method was used in the first example in Chapter 6, and the simple Euler method was used in the second example. The values of $S^{im}(t)$ used in computing the right side of (G-4) at each integration step were obtained by linear interpolation of the given values of S^{im} at discrete time instants.

The specific performance indices evaluated by the above method were the norm index in (6-6) (in the first example), and the J_N index in (6-40) (in the second example). The successful use of the PGM algorithm in the two examples in Chapter 6 indicate that the above method is a satisfactory one.

G.4 Evaluation of PGM Results using J_s

In the example discussed in section 6.3, the PGM algorithm was applied to the problem of minimizing the J_N performance index. The result of this application was a sequence of points $\{\hat{s}_i\}$ in α. It was of interest to compute the value of the J_s performance index (Skelton's probability upper bound) for each of these $\hat{s}_i$'s. This problem of evaluating $J_s(\hat{s}_i)$ given the point $\hat{s}_i$ is very similar to the evaluation problem discussed in part b) of the above section G.3. The point $\hat{s}_i$ is defined by its corresponding matrix time function $S^i(t)$, just as $\hat{s}_{im}$ was defined by $S^{im}(t)$. Also, $S^i(t)$ was available (computationally) in the form of its values at discrete instants of time over the time interval $[t_o, T]$. The intermediate values of $S^i(t)$ were approximated by linear interpolation. Because of these similarities, the same evaluation method as discussed in section G.3 was used to compute J_s, which is defined in (6-34).

The above evaluation method was used in computing the values of J_s shown in Figures 6.12 and 6.17. In these figures, the results of applying the PGM algorithm to the J_N-problem and of applying the DGIM algorithm to the J_s-problem are compared. The values of J_s shown are only approximate ones, becuase the linear interpolation approximation of $S^i(t)$ between the given time instants introduces errors into the computation of J_s. However, the DGIM results and the PGM results were both evaluated using the same approximate method, so the comparison of the results is a fair one.

G.5 Computation of Δ_i and Δ_i^o

In the example problems discussed in Chapter 6, it was required that the quantities Δ_i (defined in (6-13)) and Δ_i^o (defined in (6-14)) be computed. As explained in section 6.2.2, these numbers were measures of the "angle" between two vectors in the space σ (σ is specified in Definition 3 of Chapter 3). By the definitions in Chapter 6, we have:

$$\Delta_i = \left\| \frac{\hat{q}_i}{\|\hat{q}_i\|\sigma} - \frac{DJ(\hat{s}_i{*})}{\|DJ(\hat{s}_i{*})\|\sigma} \right\|_\sigma \tag{G-6}$$

$$\Delta_i^o = \left\| \frac{\hat{q}_i}{\|\hat{q}_i\|\sigma} - \frac{DJ(\hat{s}^o)}{\|DJ(\hat{s}^o)\|\sigma} \right\|_\sigma \tag{G-7}$$

The "vectors" $\hat{q}_i$, $DJ(\hat{s}_i{*})$, and $DJ(\hat{s}^o)$ are all given elements in the space σ, and the norm $\|\cdot\|\sigma$ is defined in equation (3-11). The computation of the norm of an element in σ is the essential problem in the determination of Δ_i and Δ_i^o.

To describe the method used to compute the norm of an element in σ, we consider a typical given element $\hat{e} = [e_F, e(t)] \in \sigma$. The vectors e_F and e are ℓ^2-dimensional, and $e(t)$ is defined on the time interval $[t_o, T]$. The norm of $\hat{e}$ is defined to be:

$$\|\hat{e}\|\sigma = \{e_F \cdot e_F + \int_{t_o}^{T} e(t) \cdot e(t)dt\}^{1/2}. \tag{G-8}$$

where the dots indicate the Euclidean inner product. As discussed in section 3.4, the vector $e(t)$ is formed by "stacking" the columns of the corresponding ℓ by ℓ matrix function of time, $E(t)$, and e_F is formed by "stacking" the columns of an ℓ by ℓ matrix E_F. So (G-8) can be rewritten in terms of E_F and $E(t)$:

$$\|\hat{e}\|\sigma = \left\{ \sum_{i,j=1}^{\ell} E_{F_{ij}}^2 + \int_{t_o}^{T} \sum_{i,j=1}^{\ell} E_{ij}^2(t)dt \right\}^{1/2}, \qquad \text{(G-9)}$$

where the subscripts indicate matrix elements.

As was discussed in sections G.2, G.3, and G.4, a matrix time function such as $E(t)$ is specified computationally by its values at discrete equidistant instants of time in $[t_o,T]$. Let these time instants be $\{t_k\}$, $k = 0,1,\ldots\bar{K}$, where $t_{\bar{K}} = T$, and let $t_{k+1}-t_k = \varepsilon$ for $k = 0,1,\ldots\bar{K}-1$. For the purposes of computing the norm of $\hat{e}$, the time interval $[t_o,T]$ was divided into $(\bar{K}-1)$ subintervals $[t_k,t_{k+1})$, where $t_o < t_1 <\ldots t_{\bar{K}} = T$. Then $E(t)$ was approximated on these subintervals by

$$E(t) = E(t_k) \quad \text{when } t\in[t_k,t_{k+1}), \qquad \text{(G-10)}$$
$$\text{for } k = 0,1,\ldots\bar{K}-1.$$

Using the above approximation for $E(t)$, equation (G-9) can be approximated by:

$$\|\hat{e}\|\sigma \cong \left\{ \sum_{i,j=1}^{\ell} E_{F_{ij}}^2 + \varepsilon \sum_{i,j=1}^{\ell} \sum_{k=0}^{\bar{K}-1} E_{ij}^2(t_k) \right\}^{1/2} \qquad \text{(G-11)}$$

The above approximation to the norm was used in computing Δ_i. For notational convenience, let $\hat{p} = DJ(\hat{S}_i{}^*)$ and $\hat{q} = \hat{q}_i$. By definition of

$$\hat{q} = [q_F, q(t)], \tag{G-12}$$

$$\hat{p} = [p_F, p(t)], \tag{G-13}$$

where q_F, q, p_F, and p are ℓ^2-dimensional vectors, and $p(t)$ and $q(t)$ are defined on $[t_o, T]$. Let the ℓ by ℓ matrices Q_F, $Q(t)$, P_F, and $P(t)$ correspond to the vectors q_F, $q(t)$, p_F, and $p(t)$, respectively. That is, q_F is formed by "stacking" the columns of Q_F, etc. Then, by using (G-11) in (G-6), it follows that

$$\Delta_i \cong [\text{TERM 1} + \text{TERM 2}]^{1/2} ,$$

where
$$\text{TERM 1} = \sum_{m,n=1}^{\ell} \left[\frac{Q_{F_{mn}}}{\|\hat{q}\|\sigma} - \frac{P_{F_{mn}}}{\|\hat{p}\|\sigma}\right]^2 ,$$

$$\text{TERM 2} = \epsilon \sum_{m,n=1}^{\ell} \sum_{k=0}^{\bar{k}-1} \left[\frac{Q_{mn}(t_k)}{\|\hat{q}\|\sigma} - \frac{P_{mn}(t_k)}{\|\hat{p}\|\sigma}\right]^2 ,$$

and where $\|\hat{q}\|\sigma$ and $\|\hat{p}\|\sigma$ are also approximated using equation (G-11). The quantity ϵ is the interval of time between the time instants at which the values of $Q(t)$ and $P(t)$ are specified computationally.

The quantity Δ_i^o was computed in a similar manner. The above method of computing Δ_i and Δ_i^o was used in the results shown in Figures 6.5, 6.6, and 6.15. In the first example, ϵ = 1 sec.; in the second example, ϵ = 5 sec. (as mentioned in section G.2).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)

School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

REPORT TITLE

ON A FUNCTION SPACE APPROACH TO A CLASS OF LINEAR STOCHASTIC OPTIMAL CONTROL PROBLEMS

DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Technical Report December, 1968

AUTHOR(S) (First name, middle initial, last name)

J.Y.S. Luh
M.P. Lukas

REPORT DATE

December, 1968

7a. TOTAL NO. OF PAGES

182

7b. NO. OF REFS

40

CONTRACT OR GRANT NO.

N 0014-67-A-0226

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

TR-EE68-36

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT

Unlimited Availability

SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

U.S. Army, Navy, and Air Force

ABSTRACT

The stochastic optimal control problem considered in this report is characterized by a dynamic system which is linear in the state and control vectors, and which is disturbed by additive Gaussian white noise. Incomplete, noisy observations of the state vector are available, and the control is required to be a linear feedback function of the estimated state vector. The components of the state vector and control vector which are of interest are lumped together in a response vector, and the performance index to be minimized is then a function of the statistics of the response vector. It is shown that a well-known stochastic control problem, whose performance index is the expected value of a quadratic form on the state and control, is a special case of the more general problem described above.

The general problem is then reformulated as a problem of minimizing a nonlinear functional on a set in a Hilbert space. In this formulation, the well-known "quadratic" problem becomes one of minimizing a linear functional on the same set in the space. Conditions are derived under which the two problems are "equivalent"; that is, the linear and non-linear functionals which specify the problems take on their minimum value at the same point in the space.

A function space algorithm of Dem'yanov is then applied to the solution of the general problem. This algorithm makes use of the known formal solution to the "quadratic" problem in the iteration procedure. In function space terms, the algorithm iteratively solves the problem of minimizing the nonlinear functional by solving a sequence of linear functional minimization problems.

DD FORM 1473 (PAGE 1)

The above approach is illustrated by two example problems. In the first example, the objective is to find a "minimum variance" control for a third-order dynamic system. In the second example, the objective is to find a control which minimizes wind-gust effects on a large, flexible launch booster. The booster dynamic and wind-gust effects are modeled by a tenth-order time-varying linear differential system. The function space approach and the algorithms developed were found to be useful in obtaining good controls for both examples.